



FACULTAD DE INFORMÁTICA
UNIVERSIDAD POLITÉCNICA DE MADRID

TESIS DE MÁSTER

MÁSTER UNIVERSITARIO EN
SOFTWARE Y SISTEMAS

**DISEÑO E IMPLEMENTACIÓN
DE UN LABORATORIO
VIRTUAL DE
BIOTECNOLOGÍA**

Autor: Diego Riofrío Luzcando
Director: Dr. Jaime Ramírez Rodríguez
Co-directora: Dra. Marta Berrocal Lobo

Julio 2012

Agradecimientos

Agradezco al Servicio de Innovación Educativa de la Universidad Politécnica de Madrid, por permitirme participar en el desarrollo del proyecto Plataforma de Experimentación para los Estudios de Ingeniería y Arquitectura.

También a Mariano Rico Almodóvar, por la formación dada acerca de Open-Simulator y por su guía en el proyecto. A Marta Berrocal Lobo, por la confianza depositada en mí para el desarrollo del laboratorio virtual de biotecnología y por la ayuda prestada para la realización de este documento. A Jaime Ramírez Rodríguez, por la guía en el desarrollo del entorno virtual y del presente trabajo.

Finalmente a las personas más importantes, mis padres que siempre me han apoyado y más estos años a la distancia, y Valeria que con su ayuda, su apoyo y principalmente sin su cariño no hubiera podido concluir con este trabajo.

Resumen

Este trabajo describe el diseño y la implementación de un laboratorio virtual de biotecnología, en el que uno o más estudiantes pueden realizar una práctica de ingeniería genética. Esta práctica consiste en modificar genéticamente un árbol (chopo) para dotarlo de una mayor resistencia a enfermedades, especialmente las producidas por hongos. Como parte de este trabajo, se ha realizado un análisis sobre plataformas existentes para la creación de entornos virtuales, tales como Second Life, Bitmanagement Software Collaborate System, OpenSimulator, Open Wonderland; además se han presentado otros trabajos similares a éste existentes en la literatura científica. En este análisis, la plataforma OpenSimulator fue seleccionada por ser gratuita y de código libre, además, al estar basada en Second Life, hereda su facilidad para la creación de mundos virtuales. Una práctica de laboratorio se modela como un protocolo o secuencia de acciones que se deben llevar a cabo. Para validar las acciones del estudiante se ha implementado un objeto, llamado tutor, que verifica la correcta realización de la práctica, y proporciona pistas sobre la siguiente acción a ejecutar. Este objeto posee una programación genérica, que se puede reutilizar en el desarrollo de otras prácticas de laboratorio a través de sencillas modificaciones en su configuración. Finalmente, en este trabajo se presentan los modelos y especificaciones para la construcción del laboratorio virtual.

Abstract

This document describes the design and implementation of a virtual biotechnology laboratory, in which one or more students can perform a genetic engineering practice. In this practice a tree is genetically modified to give it resistance to diseases, especially those produced by fungi. We study virtual worlds and analyze existing virtual environment platforms for creating virtual worlds such as Second Life, Bitmanagement Software Collaborate System, OpenSimulator, Open Wonderland; in addition to this, we have reviewed other similar works existing in the literature. As a result of our analysis, OpenSimulator was selected since it is a free and open source platform that provides an easy way of creating virtual worlds and inherited from Second Life, from which OpenSimulator is based on. A laboratory practice is modeled as a sequence of actions that must be met. In order to validate the correct flow, we have implemented an object, called tutor, which verifies the process, and provides hints on the next action to be performed. This object relies on a generic programming, which can be reused to develop other virtual laboratories through simple configuration changes. Finally, this document presents the models and specifications for the construction of the virtual laboratory.

Índice general

1. Introducción	1
2. Estado de la cuestión	5
2.1. Mundos Virtuales	5
2.2. Plataformas para el Desarrollo de Mundos Virtuales	7
2.2.1. Second Life	7
2.2.2. Bitmanagement Software Collaborate System	8
2.2.3. OpenSimulator	9
2.2.4. Open Wonderland	14
2.3. Integración del e-learning con los Mundos Virtuales	17
2.3.1. MOODLE	17
2.3.2. SLOODLE	18
2.4. Educación por medio de Mundos Virtuales	18
2.4.1. Aplicaciones de Second Life y OpenSim para la educación	20
2.4.2. Crítica de las aplicaciones educativas para Second Life y Open-Sim	24
3. Planteamiento del Problema	25
3.1. Conceptos básicos de Biotecnología	25
3.1.1. Manipulación genética	25
3.1.2. Clonado de genes	25
3.1.3. PCR (Reacción en Cadena de la Polimerasa)	26
3.1.4. Transformación de plantas	26
3.1.5. Cultivo <i>in vitro</i> de plantas	26
3.2. ¿Qué es un laboratorio de biotecnología?	27
3.3. Ejercicio práctico a modelar	30
3.3.1. Objetivos de Aprendizaje	30
3.3.2. Protocolo de la Práctica	31
3.4. Descripción del laboratorio virtual	34
3.4.1. Justificación de la solución virtual	34
3.4.2. Requisitos	35
4. Solución adoptada	39
4.1. Introducción	39

4.2. Descripción de la Arquitectura	39
4.3. Modelos 3D realizados	41
4.3.1. Edificio	41
4.3.2. Objetos	45
4.4. Modelo del Protocolo	51
4.5. Modelo de Dominio	55
4.6. Diseño de los Objetos Activos	56
4.6.1. Descripción de los Objetos Activos	56
4.6.2. Diseño del Componente Tutor	62
4.7. Diseño de la interacción entre objetos	69
5. Conclusiones y trabajos futuros	83
Bibliografía	87

Índice de figuras

2.1. Second Life	7
2.2. BS Contact	9
2.3. Hippo Viewer	10
2.4. Herramienta de los visores para SL y OS para construcción de objetos	11
2.5. Diagrama de la Arquitectura Grid de OpenSim	13
2.6. Open Wonderland	15
2.7. Enfoque en educación de SL	19
3.1. Sala de computadores en el CBGP de la UPM	28
3.2. Fitotrón en el CBGP de la UPM	28
3.3. Laboratorio en el CBGP de la UPM	29
4.1. Edificio del laboratorio	42
4.2. Vestíbulo	42
4.3. Patio	42
4.4. Sala principal del laboratorio	43
4.5. Sala de cabinas de flujo laminar	43
4.6. Sala de autoclavado	44
4.7. Fitotrón o sala de crecimiento	44
4.8. Seleccionador de práctica	45
4.9. Poyata	46
4.10. Pizarra	46
4.11. Silla	46
4.12. Libros	46
4.13. Caja de guantes de latex	47
4.14. Pecha con batas	47
4.15. Lavamanos	47
4.16. Vitrina de instrumentos	48
4.17. Vaso de precipitados	48
4.18. Bandeja de botes	48
4.19. Fregadero	49
4.20. Agitador	49
4.21. pH-metro	49
4.22. Frigorífico	50
4.23. Vitrina de productos	50

4.24. Botella para autoclavar	50
4.25. Autoclave	51
4.26. Cabina de flujo laminar	51
4.27. Esterilizador	51
4.28. Bote con planta	51
4.29. Macro procesos (Protocolo)	52
4.30. Procedimientos (Protocolo)	52
4.31. Preparación de medio de enraizamiento (Protocolo)	53
4.32. Micropropagación de material vegetal (Protocolo)	54
4.33. Modelo de Dominio	55
4.34. Especificación de dependencias del tutor	65
4.35. Diagrama de secuencia del inicio de la práctica (i)	70
4.36. Diagrama de secuencia del inicio de la práctica (ii)	71
4.37. Diagrama de secuencia del inicio de la práctica (iii)	72
4.38. Diagrama de secuencia de la creación del medio (i)	73
4.39. Diagrama de secuencia de la creación del medio (ii)	74
4.40. Diagrama de secuencia de la creación del medio (iii)	75
4.41. Diagrama de secuencia del proceso de mover el medio a botes de vidrio (i)	76
4.42. Diagrama de secuencia del proceso de mover el medio a botes de vidrio (ii)	77
4.43. Diagrama de secuencia del proceso de mover el medio a botes de vidrio (iii)	78
4.44. Diagrama de secuencia del proceso de mover el medio a botes de vidrio (iv)	79
4.45. Diagrama de secuencia del crecimiento de material vegetal (i)	80
4.46. Diagrama de secuencia del crecimiento de material vegetal (ii)	81

Índice de cuadros

2.1. Comparativa entre OpenSim y Open Wonderland.	16
4.1. Descripción del seleccionador de práctica como objeto activo	56
4.2. Descripción de la percha con batas como objeto activo	56
4.3. Descripción de la silla de la poyata como objeto activo	56
4.4. Descripción de la pizarra de la poyata como objeto activo	57
4.5. Descripción del libro de fundamentos como objeto activo	57
4.6. Descripción del libro de protocolo como objeto activo	57
4.7. Descripción de la caja de guantes como objeto activo	57
4.8. Descripción del lavamanos como objeto activo	57
4.9. Descripción de la vitrina de instrumentos como objeto activo	58
4.10. Descripción de la vitrina de productos como objeto activo	58
4.11. Descripción del Frigorífico como Objeto Activo	58
4.12. Descripción del Vaso de precipitados como Objeto Activo	59
4.13. Descripción del Fregadero como Objeto Activo	59
4.14. Descripción del Agitador como Objeto Activo	59
4.15. Descripción del pH-metro como Objeto Activo	59
4.16. Descripción de la Botella para autoclavar como objeto activo	60
4.17. Descripción del Autoclave como objeto activo	60
4.18. Descripción de la Cabina de flujo laminar como objeto activo	60
4.19. Descripción del Esterilizador como objeto activo	60
4.20. Descripción de la Bandeja de botes como objeto activo	61
4.21. Descripción del Bote con planta como objeto octivo	61
4.22. Descripción del Fitotrón como objeto activo	61
4.23. Descripción del Tutor como objeto activo	62

Capítulo 1

Introducción

La incorporación de nuevas tecnologías en el ámbito educativo ha logrado que millones de personas en todo el planeta estén conectadas intercambiando conocimientos en espacios colaborativos. El uso de estos espacios están mejorando los sistemas de educación tradicionales, pues incentivan la participación activa de los estudiantes.

Por otro lado, desde inicios de este siglo, los mundos virtuales han tenido un auge, gracias a la aparición de videojuegos como *The Sims* y *World of Warcraft*, y entornos virtuales colaborativos como *Second Life*. Este último ha permitido la creación de terrenos de diferentes tipos, como paisajes naturales, futuristas o réplicas de lugares reales como París o San Francisco. Algunos desarrolladores han utilizado los mundos virtuales para crear entornos dedicados a la educación aprovechando sus detalles gráficos, las facilidades sociales y la posibilidad de poder acceder desde cualquier lugar del mundo; estos entornos educativos se han utilizado, por ejemplo, para dictar clases o simplemente para presentar datos sobre algún tema.

Este trabajo se engloba en el campo de las aplicaciones educativas de los mundos virtuales y, el aprendizaje por medio de prácticas de laboratorio. Concretamente, se presenta el desarrollo de un laboratorio virtual de biotecnología en el que uno o más alumnos pueden realizar una práctica. Este laboratorio se desarrolló sobre *OpenSimulator*, que es una plataforma de código libre y gratuita, que proporciona muchas facilidades para desarrollar entornos virtuales multiusuario con una buena calidad a un coste relativamente bajo. Además, por estar basada en *Second Life*, posee una considerable cantidad de documentación y un gran respaldo de una activa comunidad de desarrolladores.

La virtualización de una práctica de laboratorio real en un entorno virtual tiene varios beneficios tales como eliminar los costes del material de laboratorio (entre los 10.000€ a 24.000€ por cada alumno para el caso de esta práctica); reducir el tiempo que se tarda en realizarla; evitar los riesgos a los que pueden estar expuestos los estudiantes; y tener mayor control de lo que están haciendo los estudiantes al poder obtener datos que pueden pasar desapercibidos a simple vista.

CAPÍTULO 1. INTRODUCCIÓN

El laboratorio virtual que se ha desarrollado en este trabajo consta de un edificio donde se puede realizar la práctica implementada en este trabajo así como otras prácticas de biotecnología que se planteen en el futuro. Tiene varias zonas y salas, como se puede ver en el apartado 4.3.1. Dentro de estas salas se puede encontrar el instrumental y material necesario para la realización de la práctica.

El objetivo de la práctica descrita en este trabajo es modificar genéticamente un árbol (chopo), micropropagado de otro no modificado, para que pueda presentar resistencia a enfermedades como las producidas por hongos. Ahora bien, dada la complejidad y la extensión de esta práctica, que requiere la ejecución de aproximadamente 100 acciones por parte del estudiante, este trabajo solo presenta el diseño e implementación de la primera fase de esta práctica, que supone aproximadamente un 50 % de toda la práctica. Cabe señalar que el desarrollo de un 50 % de la práctica, en realidad, representa más del 50 % de todo el proyecto, ya que muchos de los componentes que se han desarrollado para la primera fase serán aprovechados en las fases posteriores de la misma práctica, y posiblemente en otras prácticas que se planteen en el futuro.

Con el objetivo de promocionar procesos de innovación educativa que ayuden a mejorar la calidad de la educación, el Servicio de Innovación Educativa de la Universidad Politécnica de Madrid (UPM) impulsa el proyecto PEIA-UPM: Plataforma de Experimentación para los Estudios de Ingeniería y Arquitectura. El laboratorio virtual de biotecnología presentado en el presente trabajo forma parte de este gran proyecto de la UPM. Éste ha sido desarrollado por el GIE de Biotecnología conformado por biotecnólogos del Centro de Biotecnología y Genómica de Plantas (CBGP) de la UPM, e informáticos de la Facultad de Informática de la UPM y la Universidad Autónoma de Madrid, dirigidos por la Dra. Marta Berrocal Lobo, profesora de la Escuela Técnica Superior de Ingenieros de Montes.

La estructura del resto de la memoria es la siguiente:

El segundo capítulo describe el estado de la cuestión; éste contiene antecedentes, conceptos básicos y ejemplos de mundos virtuales. Además se presentan algunas plataformas en las que se pueden desarrollar entornos de este tipo. Las plataformas mencionadas son: Second Life, Bitmanagement Software Collaborate System, OpenSimulator y Open Wonderland. Asimismo, se describen algunas aplicaciones educativas desarrolladas en Second Life y OpenSimulator.

A continuación, el tercer capítulo detalla el planteamiento del problema a desarrollar, es decir, se definen los conceptos básicos de biotecnología necesarios para comprender dicho problema y proponer una solución. Asimismo, se presenta la práctica a realizar por el alumno, sus objetivos de aprendizaje y el proceso o protocolo que se debe seguir. Finalmente, se argumenta la conveniencia de un entorno virtual para este problema y se enumeran los requerimientos a satisfacer.

El cuarto capítulo explica la solución adoptada mediante la definición de la arquitectura y el diseño del entorno virtual por medio de objetos 3D que representan a los

objetos reales, así como los diagramas o tablas que explican las interacciones entre ellos y las acciones que deben realizar éstos para dar soporte a todo el protocolo de la práctica.

Dentro del cuarto capítulo se define también un objeto, que no representa a ninguno real, llamado "tutor". Éste valida las acciones que desea realizar el estudiante para verificar que la práctica se está realizando correctamente; además presenta ayuda de cómo se debe realizar la siguiente acción. Este objeto posee una programación genérica, que se puede reutilizar en el desarrollo de otras prácticas de laboratorio, no solo de biotecnología, por medio de sencillas modificaciones en su configuración.

Finalmente, el quinto capítulo enumera las conclusiones a las que ha llegado tanto por la investigación realizada sobre los entornos virtuales para la educación, como por el desarrollo de la práctica de laboratorio antes mencionada. Asimismo, este capítulo también plantea algunas líneas de trabajo futuro que se podrían derivar de este trabajo.

Capítulo 2

Estado de la cuestión

En este apartado se pretende dar una visión global acerca de la educación por medio de entornos virtuales, para esto es necesario presentar algunos conceptos sobre mundos virtuales y analizar algunas plataformas existentes para el desarrollo de éstos.

2.1. Mundos Virtuales

Los mundos virtuales nacen de los *Multi-User Dungeons* (MUD)¹. El primer MUD era basado en texto y fue programado en 1978 en lenguaje de ensamblador por Roy Trubshaw con co-autoría de Richard Bartle, quienes después crearon el *Multi-User Dungeon Definition Language* (MUDDL). Posteriormente se agregaron gráficos a éstos cambiando su nombre a *Massively-Multiplayer Online Role-Playing Games* (MMORPG).

En 1961, se crea el sistema *Programmed Logic for Automatic Teaching Operations* (PLATO), desarrollado para dar tutorías a alumnos, que operaba en un computador central donde se encontraban los programas al que accedían los estudiantes por medio de terminales. Al ser PLATO uno de los primeros sistemas para redes de computadores, se crearon varios programas que exprimían el potencial de las mismas, tanto es así, que se comienza a hablar sobre las comunidades *online*. Uno de los primeros programas creados para PLATO se llama Avatar -desarrollado en 1979- y puede ser considerado el primer mundo virtual gráfico que alcanzó un cierto éxito.

Bartle en su libro “*Designing Virtual Worlds*”[Bartle, 2003], menciona algunas convenciones relativas a los mundos virtuales, entre las que se indica que el mundo tiene reglas automatizadas que permiten a los jugadores efectuar cambios en él. Asimismo, la interacción con el mundo virtual se realiza por medio de un personaje

¹Los mundos virtuales también suelen ser llamados MUDs.

o avatar en tiempo prácticamente real (dependiendo de la velocidad de conexión), el mundo es compartido con otros jugadores y además es persistente.

Mark Bell [Bell, 2008] unifica las convenciones de Bartle con definiciones de otros autores, y describe a los mundos virtuales como “una red síncrona y persistente de personas, representadas por un avatar, facilitada por una red de computadores”. A estas personas o usuarios se los suele llamar jugadores (*players*), debido a que la gran mayoría de estos mundos son videojuegos; esto se debe a que la inversión para su desarrollo es costosa, y desde que aparecieron los gráficos por computador se han dirigido más a este tipo de software.

Los mundos virtuales se han creado con el propósito de entretener a sus usuarios. Además, al tener contacto con otros usuarios se facilitan las relaciones sociales. Bartle [Bartle, 2005] concluye que lo que hace divertido a un mundo virtual es la posibilidad que tienen las personas de encontrar quiénes son en realidad, pues dentro de éstos se pueden sentir libres de perjuicios y mostrarse como son.

En la actualidad existe un sinnúmero de mundos virtuales, los más populares son los siguientes:

- World of Warcraft². Según el libro de records Guinness es el juego *online* con más subscriptores³.
- Second Life⁴.
- The Sims⁵.
- Habbo Hotel⁶.
- EverQuest⁷.
- Rune Scape⁸.

²<http://eu.battle.net/wow/en/>

³[http://www.guinnessworldrecords.com/world-records/6000/most-popular-subscription-based-massively-multiplayer-online-role-player-game-\(mmorpg\)](http://www.guinnessworldrecords.com/world-records/6000/most-popular-subscription-based-massively-multiplayer-online-role-player-game-(mmorpg))

⁴<http://secondlife.com>

⁵<http://thesims.ea.com/>

⁶<http://www.habbo.com/>

⁷<http://everquest.com/>

⁸<http://www.runescape.com/>

2.2. Plataformas para el Desarrollo de Mundos Virtuales

2.2.1. Second Life

Second Life (SL) es un mundo virtual que consiste en un terreno dividido en regiones de tamaño fijo, donde los usuarios controlan avatares e interactúan entre ellos. Además pueden crear, modificar o eliminar objetos según los permisos que tengan los avatares en cada región.

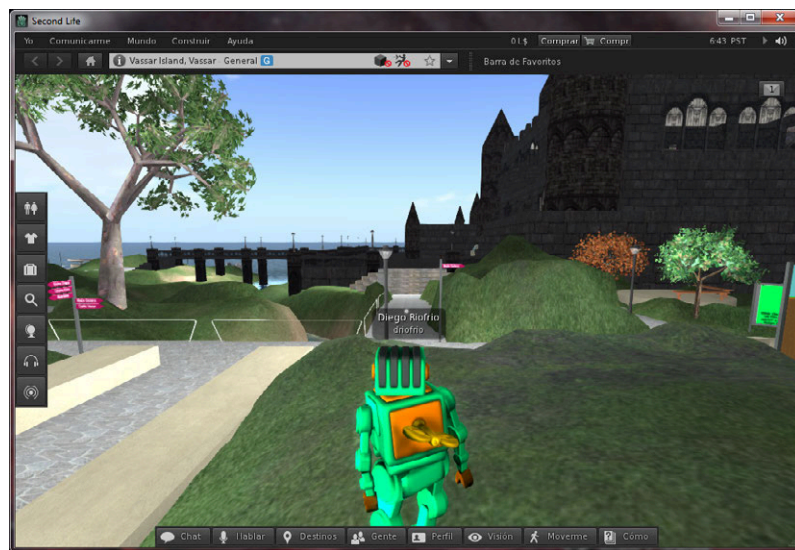


Figura 2.1: Second Life

Como se menciona en su Wiki oficial [Linden Research, Inc., 2011b], SL fue creado en 2001 como videojuego, similar a un shooter game⁹, con el nombre de LindenWorld, basándose en las ideas de Philip Rosedale (ex CEO de Linden Lab).

LindenWorld fue creado para la investigación del hardware háptico que desarrollaba la empresa. Posteriormente, dichas investigaciones fueron abandonadas para centrarse en su mundo virtual, y así en 2002 presentaron su versión beta y en 2003 lanzan SL con el lema “*Your World. Your Imagination*” (Tu Mundo. Tu Imaginación)¹⁰.

Para dotar de comportamiento a los objetos SL posee un lenguaje de *scripting* llamado *Linden Scripting Language* (LSL), que posee una sintaxis parecida al lenguaje

⁹Shooter game es un tipo de videojuego donde el usuario mediante un tipo de arma dispara a objeto u otros personajes.

¹⁰Trailer de Second Life del 2003 (http://www.youtube.com/watch?v=KHH2CAE9Y6o&feature=player_embedded)

C. Mediante este lenguaje de *script* es posible cambiar la posición de un objeto, su tamaño, color, etc.

SL posee su propia moneda llamada Linden Dollar (L\$) la cual rige el sistema económico de este mundo. En este régimen monetario Linden Lab supervisa el correcto manejo de la moneda como si fuera un gobierno central en el mundo real.

Se puede comprar L\$ con dólares americanos con un cambio especial mediante la plataforma LindeX¹¹. Aprovechando esto dentro del mundo virtual se han creado bancos virtuales que ofrecen una tasa de interés del 0.1 % diario en promedio por los ahorros en L\$.

Como parte de esta economía se pueden comprar y vender objetos como vestimenta, edificios, muebles, entre otros. Gracias a esto muchas empresas reales han creado oficinas dentro de este videojuego; de esta forma en 2006 [Linden Research, Inc., 2011b] un residente de SL llamado Anshe Chung declaró ser el primero en ganar un millón de dólares americanos gracias a la venta de bienes raíces dentro de SL.

Ernstberger [Ernstberger, 2009] explica que, debido a que SL trata de replicar la realidad, los usuarios se comportan de una forma similar a como se comportarían en su vida, por lo tanto, gracias a esto, SL podría ser una herramienta muy útil para realizar experimentos de economía.

2.2.2. Bitmanagement Software Collaborate System

BS Collaborate [Bitmanagement Software GmbH,] es una herramienta de servidor creada para aplicaciones de realidad virtual, que maneja múltiples conexiones. Permite la comunicación entre varios usuarios que manipulan objetos 3D por medio de una aplicación cliente llamada BS Contact¹², la cual es una aplicación de renderización que permite visualizar escenas 3D creadas en formato VRML o X3D.

El servidor maneja la seguridad de acceso utilizando la base de datos de usuarios conectada a él, y además soporta otras funcionalidades como posiciones de los avatares y envío de mensajes entre usuarios.

Tanto el servidor como la aplicación cliente son vendidos por una empresa alemana llamada Bitmanagement Software GmbH, aunque de forma gratuita permite descargar una versión *trial* de sus productos.

A diferencia de SL, BS Collaborate se puede instalar en un servidor propio, pero al no ser una aplicación de distribución libre es necesario pagar una licencia de 19.990 € por servidor. Además se paga una licencia distinta por BS Contact dependiendo del número de clientes, es decir, el precio por uno solo es de 300 €, pero por cinco

¹¹<https://secondlife.com/my/lindex/market.php> se necesita cuenta para poder ingresar

¹²<http://www.bitmanagement.de/en/products/interactive-3d-clients/bs-contact>

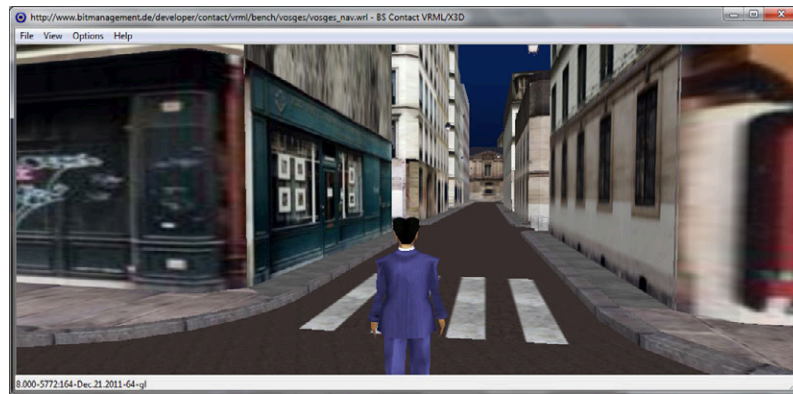


Figura 2.2: BS Contact

licencias el coste es de 260 € por cada una. En su portal web se puede ver estos precios con más detalle¹³.

2.2.3. OpenSimulator

OpenSimulator, o también llamado OpenSim (OS), es un sistema 3D de código abierto, multi-plataforma y multi-usuario [OpenSim Team, 2011b]. El proyecto fue fundado en 2007 por Darren Guard, como se menciona en la historia de OS [OpenSim Team, 2011a]. En enero de 2007, SL lanzó un nuevo cliente de código abierto además de una librería para que otros desarrolladores puedan crear nuevos clientes gráficos que se conecten a su servidor. Gracias a aquello nace la idea de OS, como un servicio 3D al que se pueda conectar dicho cliente gráfico de SL.

Está escrito en C# y por lo tanto corre en sistemas operativos Windows sobre el *.NET Framework* de Microsoft, y sobre el *framework* Mono para sistemas operativos Linux. En la actualidad, se encuentra en la versión 0.7 considerada todavía como una versión alfa, puesto que los desarrolladores están todavía corrigiendo errores y así estabilizándola antes de lanzar su versión 1.0.

OS permite recrear mundos de la misma forma que en SL, utilizando las mismas primitivas y su mismo lenguaje *script* (LSL) para dotar de comportamiento a los objetos. Además posee algunas funciones propias que se pueden utilizar dentro de los *scripts* de la misma forma que las de LSL, denominadas funciones OSSSL, pues tienen el prefijo *os* en vez del *ll* que tienen las de LSL. Su API está bien documentada en el portal web de OS ¹⁴.

Dentro del mundo virtual un avatar se mueve por regiones donde puede interactuar con objetos, construir nuevos objetos y editar sus propiedades, siempre que

¹³<http://www.bitmanagement.de/en/shop/pricing>

¹⁴http://opensimulator.org/wiki/OSSL_Implemented

CAPÍTULO 2. ESTADO DE LA CUESTIÓN

disponga de los permisos apropiados. Una región es un área de terreno que tiene 256x256 metros de extensión, de acuerdo con el diseño original de SL.

Para poder acceder al mundo virtual se necesita utilizar una aplicación cliente, también denominada visor. Por medio de ésta se inicia sesión y después de recibir la respuesta del servidor con la última posición del avatar y con la información de los objetos que se encuentran alrededor renderiza todo esto. Es decir, esta aplicación es la que se encarga de la visualización 3D. Todos estos visores permiten la comunicación entre avatares, ya sea por medio de chat o por voz. Los visores reciben el texto o el sonido y se lo envían al servidor para que los redirija al visor de los otros usuarios. Existen varios visores que pueden conectarse tanto a SL como a OS, entre estos cabe destacar:

- Hippo Viewer¹⁵.
- Imprudence¹⁶.
- Phoenix Viewer¹⁷.
- RealXtend¹⁸.
- Singularity¹⁹.
- Second Life Viewer²⁰.

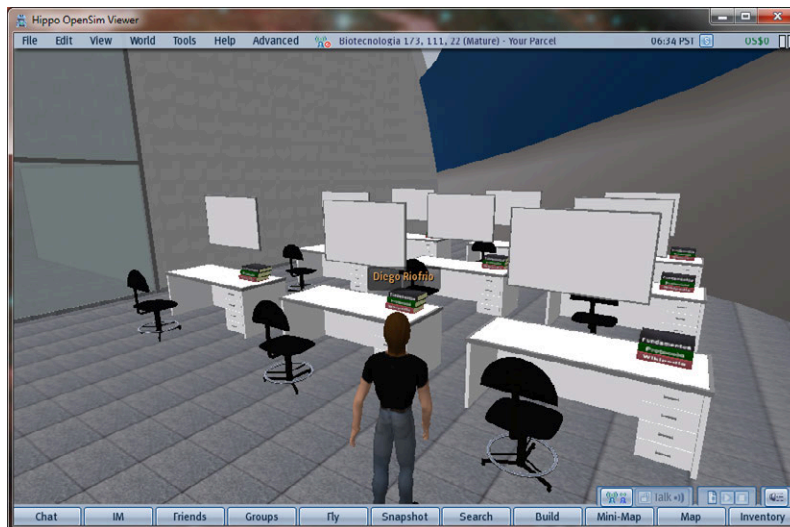


Figura 2.3: Hippo Viewer

¹⁵<http://sourceforge.net/projects/opensim-viewer/>

¹⁶<http://blog.kokuaviewer.org/>

¹⁷<http://www.phoenixviewer.com/>

¹⁸<http://www.realxtend.org/>

¹⁹<http://www.singularityviewer.org/>

²⁰<http://secondlife.com/support/downloads/>

Todos estos visores permiten crear objetos 3D a base de primitivas básicas las cuales son: cubo, prisma, pirámide, tetraedro, cilindro, semicilindro, cono, semicono, esfera, semiesfera, toroide, tubo, anillo, árbol y hierba. En la figura 2.4 se muestra la herramienta con la que se crean las primitivas en el mundo. Además con ésta se pueden cambiar propiedades de los objetos tales como: tamaño, ubicación, color, textura, rotación, etc. Para crear objetos 3D más complejos como un edificio o una mesa, se pueden unir varias primitivas con diferentes dimensiones. Por ejemplo, la mesa se puede crear con cinco cubos, a los que se les cambia de tamaño alargando y estrechando sus lados para finalmente unirlos. OS también permite importar modelos 3D creados por otras aplicaciones en formato Collada a los cuales se los llama *mesh*.

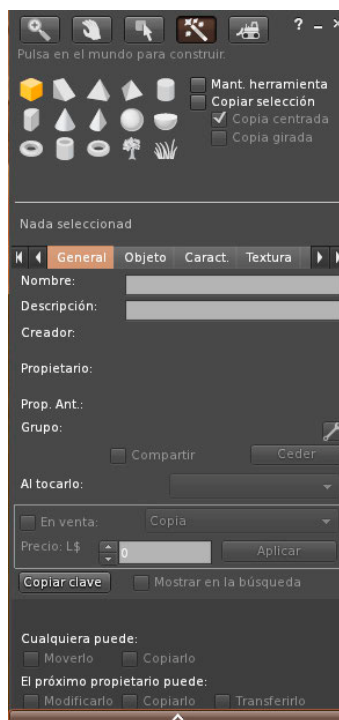


Figura 2.4: Herramienta de los visores de SL y OS para construcción de objetos

Por medio de la herramienta de construcción de la figura 2.4 también se puede dar comportamiento a los objetos mediante *scripts*, los cuales se incluyen en el inventario de la primitiva. Estos *scripts* deben ser escritos en el mismo lenguaje que utiliza SL (LSL), que es un lenguaje de programación estructurado dirigido por eventos con una sintaxis similar a la de C.

Un *script* programado en LSL se puede dividir en estados, para definir de mejor forma los cambios de un objeto y además permitir que un objeto se comporte de diferente manera en cada uno de ellos. Por ejemplo, una pelota que inicialmente se encuentra en un estado de reposo, al tocarla puede cambiar a un estado de movimiento, y después al volver a tocarla puede regresar a su estado inicial. Todo *script* comienza con un estado llamado *default* y para crear uno nuevo se debe anteponer la palabra *state* seguido de un espacio y el nombre que se desee dar a éste.

CAPÍTULO 2. ESTADO DE LA CUESTIÓN

Como ya se mencionó, este lenguaje está dirigido por eventos, es decir, cuando existe algún cambio en el entorno, el script lo detecta y realiza los pasos definidos para dicho evento. Existen varios tipos de eventos definidos en el portal de SL²¹; entre éstos algunos de los más utilizados son:

- *state_entry*: es el evento que se ejecuta al iniciar el estado.
- *touch*: se produce cuando el usuario toca con el puntero del ratón el objeto.
- *listen*: se dispara cuando recibe un mensaje por un canal de chat desde otro objeto o desde el panel de mensajería del visor.
- *timer*: se ejecuta cada cierto tiempo, configurado por medio de la función *llSetTimerEvent*.
- *sensor*: se produce cuando detecta que un objeto o avatar entra en su campo de acción definido por la función *llSensor* o *llSensorRepeat*.

LSL tiene varios tipos de datos, la mayoría similares a los de los demás lenguajes que son: flotante (*float*), entero (*integer*) y cadena de caracteres (*string*). Existen otro tipos de datos propios de LSL como:

- *vector*: es un tipo de dato que contiene tres números flotantes y que se utiliza para definir posiciones de los objetos en X, Y y Z.
- *rotation*: define un ángulo de orientación del objeto 3D.
- *key*: sirve para definir a un identificador único de todo lo que se encuentra en el entorno.
- *list*: es un tipo de dato especial que puede contener cero o más elementos de cualquier tipo de dato, similar a un *ArrayList* en Java.

De igual forma, permite la utilización de operadores para realizar operaciones como: suma, resta, multiplicación, división, módulo, incremento, decremento, comparaciones y operaciones binarias, etc.

Como todo lenguaje posee funciones propias para realizar alguna acción ya sea sobre un objeto, el entorno, o un avatar. Éstas también se encuentran definidas en su portal web²² y es importante verificar su estado de implementación en OS²³ antes de utilizarla por primera vez, para así evitar problemas.

En OS también existen otras funciones, que no se encuentran implementadas en SL, llamadas OSSL²⁴, éstas también se pueden utilizar dentro de los *scripts* y realizan acciones que las funciones de SL no permiten o necesitan de más líneas de código para realizar lo mismo. Para utilizar una de estas funciones por primera vez

²¹http://wiki.secondlife.com/wiki/Category:LSL_Events

²²http://wiki.secondlife.com/wiki/Category:LSL_Functions

²³http://opensimulator.org/wiki/LSL_Status/Functions

²⁴http://opensimulator.org/wiki/OSSL_Implemented

dentro del entorno es necesario modificar sus permisos de ejecución en el archivo OpenSim.ini y reiniciar OS.

Arquitectura de OpenSim

OS tiene dos formas de operar, la primera es llamada modo *StandAlone*, en la que todas las regiones se encuentran en un mismo servidor. La segunda se llama modo *Grid*, y es totalmente distinta a la anterior, pues en ésta las regiones se encuentran en varios servidores que ejecutan OS y se encuentran conectados por medio de una red o por medio de Internet como *New World Grid*²⁵, cuyo portal web permite conectar una región de OS a todo su *grid* de regiones.

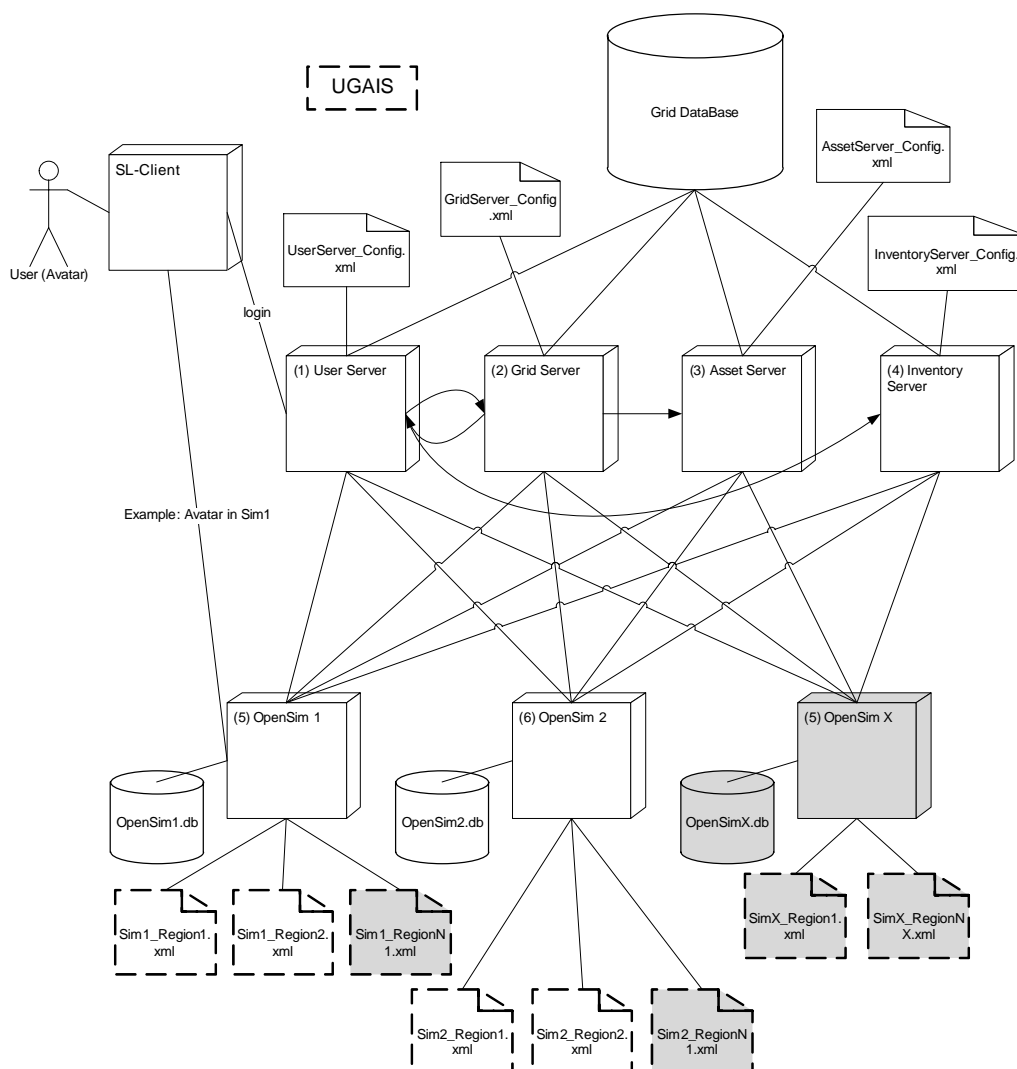


Figura 2.5: Diagrama de la Arquitectura Grid de OpenSim²⁶.

²⁵<http://www.newworldgrid.com/>

Desde el punto de vista de la arquitectura del software, cada región proporciona cinco servicios principales conocidos por el acrónimo UGAIM, como se muestra en la Figura 2.5 (*User, Grid, Asset, Inventory, Messaging*). El *UserServer* es el servicio que autentifica a los usuarios para que puedan ingresar al grid por medio de su avatar, cada avatar está definido por un Identificador Único Universal o UUID (*Universally Unique Identifier*) -por sus siglas en inglés-. El *GridServer* autentifica cualquier otra cosa en el grid que no sea un avatar. El *AssetServer* administra todos los objetos (*Assets*) que existen como, sonidos, texturas, imágenes, notas, scripts u objetos, a los cuales les otorga un UUID para que puedan diferenciarse. *InventoryServer* administra los inventarios de los avatares y de los objetos, es decir, cada objeto y cada avatar posee un contenedor o inventario que puede albergar más objetos, aquí es donde interviene el *InventoryServer* que enlaza el UUID del objeto o avatar con todos los objetos de su inventario. Finalmente el *MessagingServer* administra la comunicación por medio de mensajes entre avatares manteniendo logs de dichos mensajes.

Cuando se configura OS para que se ejecute en modo *StandAlone*, la región se provee de sus propios servicios UGAIM que se ejecutan como un mismo proceso. En cambio, en el modo *Grid* estos servicios se ejecutan cada uno como un proceso individual y pueden estar corriendo en diferentes computadores.

OS tiene además tres formas de almacenar sus datos. La primera, por medio de una base de datos SQLite, la segunda es por medio de MySQL, y la tercera es mediante la utilización de Microsoft SQL Server.

Las configuraciones del tipo de arquitectura, el tipo de base de datos, permisos de funciones OSSL, así como la activación de otros módulos, se realiza en un archivo de texto llamado *Opensim.ini* que se encuentra en la carpeta *bin* de OS.

2.2.4. Open Wonderland

Open Wonderland (OW) es una herramienta de código abierto, desarrollada en Java para crear mundos virtuales en tres dimensiones. Este proyecto comenzó en 2007 en Sun Microsystems y desde 2010 está a cargo de una comunidad de desarrolladores de código abierto. Su objetivo principal es habilitar una nueva forma de 3D para la Web, para que en un futuro los navegadores permitan a los usuarios visualizar y navegar por un mundo virtual tridimensional.

Kaplan y Yankelovich [Kaplan and Yankelovich, 2011] indican que OW, de igual forma que otras herramientas para entornos virtuales, utiliza un modelo cliente-servidor, con dos tipos de cliente: una aplicación web que se utiliza para la administración del servidor, y una aplicación de escritorio desarrollada en Java, que se utiliza para acceder y visualizar el entorno tridimensional.

²⁶Véase http://opensimulator.org/wiki/Grid_Architecture_Diagram

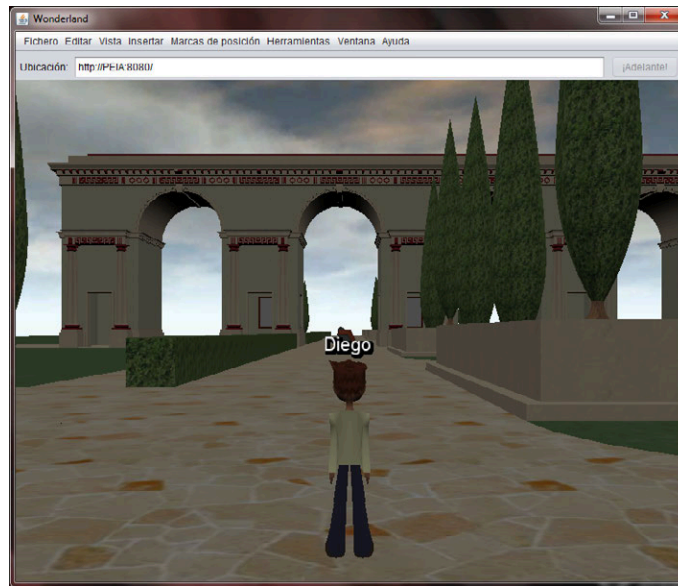


Figura 2.6: Open Wonderland

La aplicación web y el cliente de escritorio se comunican por medio de servicios web con el servidor para la autenticación y la descarga de modelos 3D y texturas, pero solo el cliente de escritorio utiliza TCP (*Transmission Control Protocol*) para obtener propiedades de los objetos y sus posiciones, además de SIP (*Session Initiation Protocol*) o RTP (*Real-time Transport Protocol*) para audio y *multimedia streaming* para vídeo.

Para la renderización se usa un motor gráfico llamado *JMonkeyEngine*²⁷, utilizado para desarrollar aplicaciones Java que necesitan realizar gráficos con *OpenGL*, pero como este motor se ejecuta en un solo proceso, se creó el motor *MT Game* como un subproyecto de OW para agregarle multiprocesamiento. OW utiliza una arquitectura modular basada en componentes Java, de código abierto, para que el entorno sea fácilmente extensible. Asimismo, se integra con estándares abiertos como *Collaborative Design Activity* (Collada²⁸) para gráficos y SIP para audio. Cada uno de estos módulos, ya sean modelos 3D o contenedores de medios, tienen que ser programados en Java para importarlos dentro del entorno; además, se promueve la publicación de estos contenidos en el portal web.

Comparativa entre OpenSim y Open Wonderland

Analizando las características de OS y OW -no se toma en cuenta SL ni Bitmanagement al no ser plataformas gratuitas- se puede llevar a cabo una pequeña comparativa como la siguiente:

²⁷<http://jmonkeyengine.com>

²⁸<http://collada.org>

Cuadro 2.1: Comparativa entre OS y OW.

Característica	OpenSim	Open Wonderland
Tipo de Licencia	Gratuita	Gratuita
Filosofía de desarrollo	Código Abierto	Código Abierto
Sistema Operativo	Multi-plataforma	Multi-plataforma
Plataforma	32 y 64 bits	32 bits
Lenguaje de desarrollo	C#	Java
Tiempo de inicio	2 minutos aprox.	10 minutos aprox.
Visor	Visores SL	Visor propio
Creación de objetos 3D	Primitivas básicas.	Programados en Java.
Importación de modelos 3D	Collada	Collada.

La diferencia de la calidad de los gráficos entre estas dos plataformas depende de la forma en que se han construido los objetos, ya que si se los ha importado como Collada no existiría diferencia, pero si se hace esto para objetos grandes como edificios, la aplicación cliente demorará en dibujarlos. Para solucionar este problema se deben utilizar las propias herramientas de construcción que otorga la plataforma, en este caso pueden existir diferencias en la calidad de los gráficos, debido a que programar los modelos en Java -como se debe hacer en OW- es más complicado que construirlos por medio de primitivas básicas, por esto al programarlos se tiende a hacer simplificaciones de diseño, que repercuten en la calidad.

OW tiene incorporado un servicio web con un portal de administración del entorno, como la creación de grupos de usuarios, la instalación o eliminación de módulos, entre otros. OS no posee por defecto un portal parecido, pero existen varios módulos desarrollados por la comunidad como Wifi²⁹ que es provisto por Diva Canto de Metaverse Ink, que permite crear una cuenta a un usuario que no la posea; además de visualizar su inventario y eliminar ítems, también proporciona al administrador la opción de eliminar usuarios, entre otras cosas.

OS tiene compatibilidad con visores para SL, y a través de éstos se pueden exportar e importar objetos que se encuentren en cualquiera de estas dos plataformas. Por otra parte, OW necesita un cliente configurado específicamente para cada servidor, que se descarga desde el portal web del mismo. Este cliente al ejecutarse necesita descargarse nuevos componentes desde la dirección IP del servidor, pero si el servicio no se encuentra disponible, el cliente no se ejecuta, aunque en la pantalla de inicio de sesión se permita cambiar de dirección del servidor.

OW permite arrastrar desde el explorador del sistema operativo y soltar dentro del visor algunos tipos de archivos, como imágenes, y así visualizarlas dentro del entorno, mientras que en OS primero hay que importarlas y después se deben incluir como texturas de una primitiva o introducirlas como archivo multimedia de una parcela. Un tipo de archivo que se puede insertar de esta forma en OW es el PDF, y este

²⁹<http://opensimulator.org/wiki/Wifi>

visor permite navegar por las páginas del archivo, mientras que para poder abrir un archivo de este tipo en OS se necesita exportar sus páginas a imágenes y después incluirlas como texturas.

Algo digno de destacar de la comunidad de OW, es su política de acceso a los modelos 3D creados por los desarrolladores, los cuales se encuentran disponibles como una librería en el portal web de OW³⁰, desde donde se pueden descargar de forma gratuita.

2.3. Integración del e-learning con los Mundos Virtuales

2.3.1. MOODLE

Gracias a la evolución de la tecnología web y la proliferación del uso de Internet se han ido creando sistemas que ayudan en la educación. Así nace el *e-learning* para proporcionar formación personalizada y a distancia.

Una de las herramientas de *e-learning* más populares se llama Moodle³¹, creada por Martin Dougiamas y lanzada en 2002. Es una aplicación web gratuita que ayuda a crear y administrar sitios web dinámicos de aprendizaje colaborativo donde los profesores pueden crear sitios para sus asignaturas, y así poder publicar material y noticias de la asignatura, o crear tareas y exámenes que los alumnos pueden realizar desde su casa. Moodle es un ejemplo de sistema de educación asíncrona, por ejemplo, el profesor crea un examen, pasado cierto tiempo el estudiante lo realiza, puede obtener o no su calificación en ese momento dependiendo de la configuración del examen, pero el *feedback* del profesor lo recibirá después de que éste revise la evaluación.

Según Kumar *et al.* [Kumar et al., 2011], la arquitectura de Moodle está compuesta por módulos independientes que se pueden reunir en seis grupos diferentes de acuerdo a su propósito: módulos de comunicación, módulos de productividad, módulos de participación de los estudiantes, módulos de administración, módulos de entrega del curso y módulos de diseño de currículum. Kumar *et al.* [Kumar et al., 2011] realizan una comparativa con otros sistemas de *e-learning* en cuanto a la arquitectura y características técnicas, y llegan a la conclusión de que el mejor es Moodle, a pesar de sus defectos de seguridad.

³⁰<http://openwonderland.org/module-warehouse/module-warehouse>

³¹<http://moodle.org>

2.3.2. SLOODLE

Es un proyecto de código abierto que permite vincular entornos virtuales en SL u OS con Moodle [SJSU's School of Library & Information Science, 2011a]. Posee algunos objetos en 3D que se pueden incluir en los entornos virtuales de enseñanza; estos objetos se comunican por medio del protocolo HTTP con el componente web de Sloodle que es el que se encarga de realizar directamente los cambios en la base de datos de Moodle. Cabe señalar que los servicios web de Moodle y Sloodle deben estar instalados en el mismo servidor.

Existen varios objetos en Sloodle [SJSU's School of Library & Information Science, 2011b], entre estos: *Registration Booth*, que permite enlazar el avatar del entorno con el usuario de Moodle; *LoginZone*, que permite delimitar zonas en las que pueden ingresar solamente avatares autenticados previamente con Moodle; *Quiz Chair*, mediante el cual los alumnos pueden realizar exámenes de selección múltiple creados en Moodle; *Prim Drop*, que se puede utilizar para que los alumnos entreguen tareas como objetos del entorno; etc.

2.4. Educación por medio de Mundos Virtuales

Con la aparición de la Web 2.0 se ha visto que la interacción entre las personas es sumamente relevante en los sistemas web, y ésta no debe ser la excepción en los sistemas de aprendizaje *online*, que deben permitir un aprendizaje colaborativo entre estudiantes.

La comunicación entre el alumno y el sistema de educación *online* pueden ser de una forma síncrona o asíncrona. Hrastinski en su tesis doctoral [Hrastinski, 2007] explica que la participación *online* de un estudiante se puede confrontar de diferentes vías, puesto que cada forma de comunicación tiene diferentes características. Una comunicación síncrona puede incrementar la motivación del alumno y reducir los errores debido a que se tiene un *feedback* inmediato; mientras que una comunicación asíncrona incrementa el esfuerzo cognitivo del estudiante, pues éste tiene más tiempo para estudiar y pensar en el tema. Finalmente sugiere que la comunicación síncrona se puede utilizar de apoyo a la asíncrona para así conseguir una mayor participación personal del estudiante.

Hoy en día SL es una de las plataformas más populares para la implementación de entornos virtuales educativos. Aquí se han creado muchos ambientes educativos: colegios, universidades y organizaciones de todo el mundo [Linden Research, Inc., 2011d], como la Universidad de California Irvine, la Academia Internacional de la Universidad de Essex, la *American Library Association*, entre otros³². Gracias a este

³²http://wiki.secondlife.com/wiki/Second_Life_Education_Directory



Figura 2.7: Enfoque en educación de SL³³.

crecimiento, Linden Lab por medio de la página web de SL da información de cómo crear un entorno virtual y qué beneficios tiene para la educación.

El folleto informativo de educación en SL [Linden Research, Inc., 2011c] promueve este tipo de aprendizaje no solo para estudiantes que se encuentren dispersos geográficamente, sino también para otros que pudiendo reunirse periódicamente necesiten flexibilidad en el horario o trabajar de forma asíncrona.

Linden Research destaca algunos beneficios de SL como plataforma para la educación *online*. Son de la opinión de que con SL se puede aumentar la atención y la participación de los alumnos, pues fue creado como una red social para fomentar las interacciones entre las personas. Además, los profesores pueden percatarse cuando un alumno se encuentra haciendo otra cosa en su computador o no se encuentra en él, debido a que pasado un tiempo de inactividad, el avatar se pone en estado adormitado.

Otros beneficios son los lugares y recursos educativos que ya se encuentran contruidos en SL como la Capilla Sixtina o las Pirámides de Egipto, así como el Museo de Innovación Tecnológica San José o la isla de la NASA, entre otros, lo que puede ser más productivo para un estudiante que verlos en un libro. Al tener varias instituciones educativas dentro de SL, se puede contactar con profesores que estén dispuestos a colaborar y al intercambio de medios de aprendizaje, ayudados por las herramientas que posee SL como su lista de correos electrónicos.

Asimismo, se puede acceder a un mayor número de estudiantes, por ejemplo, se

³³http://wiki.secondlife.com/wiki/Second_Life_Education

puede llegar a alumnos de otros países o continentes, o personas con discapacidades que no pueden asistir a clases tradicionales por la falta de accesibilidad en los campus. Además, es una forma de aprendizaje ecológica, pues a pesar de que necesita electricidad para la alimentación de la computadora del usuario y de los servidores de Linden Lab, esto representa un impacto ambiental mínimo en comparación con el uso de papel o de la contaminación emitida por los medios de transporte que utilizan los alumnos para el traslado a los centros de estudio.

Un caso de éxito [Linden Research, Inc., 2011a] es la *Open University*³⁴, cuyo centro de estudio es el más grande del Reino Unido y el único que provee de educación a distancia a través de medios innovadores. Tiene alrededor de 150 mil alumnos de grado y 30 mil de postgrado, y actualmente posee un campus³⁵ dentro de SL para ofrecer una nueva forma de educación a sus alumnos, además de los soportes que ya poseía como correo postal o teléfono.

Gracias a este campus los profesores no necesitan viajar al edificio real y triplican el número de alumnos a los que imparten docencia. Después de un tiempo, los administradores del campus observaron que los estudiantes comenzaban a sentir la necesidad de poseer un lugar en donde puedan vivir sus avatares. Para esto construyeron un edificio de residencias donde se alquilan viviendas gratuitas a corto plazo; éstas han tenido tal acogida que existe una larga lista de espera.

Al ver el éxito que tenía el campus virtual, el departamento de recursos humanos de la universidad llegó a la conclusión de que también podía utilizar SL para capacitar a sus docentes, lo cual ha tenido buena acogida.

2.4.1. Aplicaciones de Second Life y OpenSim para la educación

Para la educación por medio de entornos virtuales muchos autores han utilizado SL gracias a que mucho trabajo de programación, relacionado con los avatares y el entorno, ya se encuentra realizado. En la actualidad algunos de estos estudios están utilizando OS debido a los costos que implica realizarlos en SL, cuyas tarifas fluctúan entre 9,95 USD al mes por el uso de una parcela de 512 m^2 con aproximadamente 117 primitivas, hasta 195 USD mensual por 65.536 m^2 con 15.000 primitivas. Además, al no poder restringir el acceso a la parcela, puede acceder a ella cualquier usuario de SL, aunque se puede comprar una isla privada de 65.536 m^2 con el costo de 1.000 USD más una cuota mensual de 295 USD.

Alexandra Petrakou [Petrakou, 2010], para su estudio creó un campus virtual en SL para enseñar el idioma inglés, en el que tuvo estudiantes de Suiza y Estados Unidos. Además en el curso participaba un profesor que cumplía funciones de guía en las

³⁴<http://www.open.ac.uk/>

³⁵<http://slurl.com/secondlife/OpenUniversity/104/37/25>

lecturas y de coordinador de los grupos de discusión. También existía un observador que participaba en todas las sesiones para poder estudiar los comportamientos de los estudiantes.

En el lanzamiento del curso existieron varios problemas, por ejemplo, los estudiantes que nunca han utilizado un mundo virtual no sabían cómo navegar por él y chocaban contra otros avatares. También se presentaron problemas técnicos como configuraciones de auriculares y micrófono, o problemas de coordinación como el protocolo para hablar (cuándo hablar y cuándo callar). Además, la relación entre distancia y voz no era fácil de manejar, el volumen de la voz podía dar a entender que la persona está más cerca de lo que está su avatar. Muchos de estos inconvenientes dieron lugar a interrupciones en los debates del curso.

Al conectarse desde su visor los estudiantes aparecían en un mismo punto y luego entraban a la sala de lectura, que era similar a cualquier aula de clase, con paneles de presentaciones, sillas para los avatares de los alumnos y el docente en frente. Después de las lecturas los alumnos debían hacer presentaciones, algunos ejercicios o discusiones en grupo.

A los estudiantes se les preguntó sobre la diferencia entre hacer presentaciones en la vida real y hacerlas en el mundo virtual, a lo que contestaron que en el mundo virtual es difícil de interpretar las reacciones de la audiencia, como el lenguaje corporal, para poder modificar la forma en la que se está realizando la presentación.

Petrakou también observó que la comunicación síncrona en los sistemas de educación no es solo importante para la relación entre el docente y los alumnos, sino también para la relación entre los mismos alumnos, pues las interacciones en el mundo virtual pueden ser tan valiosas como las que se dan en un campus universitario normal. Así se vio que el propio uso del entorno virtual facilita un aprendizaje colaborativo, ya que ayuda en la socialización e interacción de los estudiantes desde el primer día, pues si el uso del mundo virtual es desconocido por los estudiantes, el aprendizaje de éste se convierte en una actividad colaborativa.

De este estudio de Petrakou se puede concluir que cuando los alumnos empiezan a acostumbrarse a utilizar los avatares e interrelacionarse por medio de ellos, es decir, cuando comienzan a no interactuar con las computadoras, sino a interactuar entre sí por medio de la computadora, en ese momento se podrá ver el verdadero potencial de los entornos virtuales.

Boulos *et al.* [Boulos et al., 2007] muestran el potencial de SL en la pedagogía mediante algunos ejemplos de educación en medicina y salud en general. El primero es un juego llamado *Nutrition Game* creado por la Universidad de Ohio, que permite aprender acerca del impacto de la comida rápida en la salud. Para ello los jugadores pueden escoger varios tipos de comida y cuanto más saludable sea la escogida, mejor será el puntaje obtenido.

El segundo ejemplo mostrado en [Boulos et al., 2007], llamado *The Heart Murmur*,

fue diseñado para la *San Jose State University*, y permite el entrenamiento en la auscultación cardíaca ³⁶. En este entorno virtual, el estudiante realiza un tour por una clínica virtual para que puedan identificar diferentes sonidos del corazón.

Otro ejemplo presentado en [Boulos et al., 2007] es la *Healthinfo Island*, desarrollada con financiación de la *National Library of Medicine* de los Estados Unidos para proveer información sobre salud. Esta isla está a cargo de un grupo de bibliotecarios y profesionales de la información de la salud, con la ayuda de epidemiólogos, médicos internistas, farmacéuticos, entre otros. En esta isla se encuentran contruidos cuatro edificios, el primero posee información médica presentada en diapositivas; el segundo tiene forma de estudio de un hogar y entrega notas al avatar con información sobre salud para el consumidor; el tercero tiene información sobre salud y bienestar, de igual forma presentada en diapositivas; mientras que el último edificio es un laboratorio para la experimentación de las diferentes formas de presentar información sobre salud que se pueden crear dentro de SL.

El último ejemplo presentado por Boulos *et al.*, se llama *The Virtual Neurological Education Centre*, creado en la *University of Plymouth* del Reino Unido. Éste muestra información sobre medicina neurológica por medio de diapositivas o de vídeos, también indica como pueden ser los síntomas de una enfermedad de este tipo mostrándolas mediante animaciones del avatar. Además, posee un aula con material y equipamiento que se utiliza en un departamento de neurología de un hospital con el que se puede interactuar.

De estos ejemplos Boulos *et al.* resaltan algunos aspectos, como la utilización de avatares que no solo pueden comunicarse entre ellos, sino que también pueden moverse con lo cual se pueden mostrar imágenes animadas de los estudiantes. También destaca el hecho de que al ser un ambiente simulado, los estudiantes de medicina pueden mejorar sus habilidades sin poner en riesgo a pacientes reales. Con ello sacan la conclusión de que realizar educación médica por medio de entornos virtuales no es imposible, pues éstos tienen un gran potencial, pero hace falta aún más investigación para poder utilizarlos en la enseñanza diaria, por ejemplo, en la definición de mejores prácticas, así como también en las formas de evitar las trampas que pudiesen cometer los alumnos.

Rico *et al.* [Rico et al., 2011] presentan una plataforma educativa dentro de un mundo virtual llamada V-LeaF (*Virtual LEArning platForm*) y desarrollada en OS, que ofrece un curso de programación en LSL para estudiantes de educación secundaria, motivado por el elevado porcentaje de alumnos españoles que abandonan la carrera de Ciencias de la Computación. Con este entorno virtual, los profesores de computación pueden enseñar programación básica de una forma dinámica, pues la mayoría de los cursos de informática tratan sobre cómo utilizar aplicaciones de

³⁶“Aplicar el oído a la pared torácica o abdominal, con instrumentos adecuados o sin ellos, a fin de explorar los sonidos o ruidos normales o patológicos producidos en los órganos que las cavidades del pecho o vientre contienen.” Tomado del diccionario de la Real Academia Española, véase en <http://buscon.rae.es/draeI/SrvltGUIBusUsual?LEMA=auscultar>

usuario, y por lo tanto no enseñan la verdadera naturaleza de la informática, esto provoca que los estudiantes tengan una idea equivocada acerca de ésta.

Al curso se le llamó “Taller de programación en Mundos Virtuales”, y trata de dar un mayor atractivo a la programación para los estudiantes de secundaria mediante algunos ejemplos que sean visualmente agradables. Por ejemplo, un bucle se puede ilustrar como una esfera que aumenta de tamaño un cierto número de veces.

Los componentes de esta plataforma son un portal web, que proporciona acceso a la documentación y administración de usuarios, un entorno virtual construido en un grid sobre OS, y otro tipo de software para *data mining*.

Para analizar la experiencia que tienen los estudiantes y los profesores dentro de V-Leaf, se creó un experimento con alumnos (entre 16 a 18 años) y docentes de dos institutos técnicos de secundaria. Al terminar el curso los dos grupos rellenaron un cuestionario con sus datos, sus experiencias con mundos virtuales, sus destrezas en programación, y sobre el entorno de V-Leaf. En este cuestionario se observó que los estudiantes ya tenían experiencia con mundo virtuales como *The Sims*, mientras que los profesores no. Los estudiantes resaltaron el aprendizaje de LSL, al que calificaron como muy interesante, gracias a que podían ver los efectos de forma gráfica (como si estuvieran jugando). Asimismo, los alumnos dijeron que la información en Internet sobre este lenguaje, y en general sobre entornos virtuales, es muy técnica y poco comprensible para ellos, mientras que los profesores observaron un alto nivel de atención e interés en sus alumnos.

En la Universidad de Leicester existe actualmente un proyecto llamado *Second World Immersive Future Teching* (SWIFT) [Beyond Distance Research Alliance and GENIE, 2011], que está siendo desarrollado en colaboración entre los departamentos *Beyond Distance Research Alliance* y *Genetics Education Networking for Innovation & Excellence* (GENIE). El proyecto se centra en las ciencias biomédicas, pero esperan que los resultados puedan ser utilizados en varios campos de la ciencia. Para esto pretenden investigar el impacto que provoca en los estudiantes la utilización de esta herramienta en el desarrollo del laboratorio.

Actualmente, se encuentran en desarrollo tres laboratorios, en el primero, los estudiantes pueden utilizar el instrumental y maquinaria de un laboratorio de genética, tomando las debidas precauciones de seguridad; el segundo laboratorio les enseña cómo se deben utilizar estos equipos para cumplir un cierto objetivo, por medio de casos de estudio; el último -que aún no se encuentra habilitado- les permitirá trabajar en grupos para realizar experimentos simulados.

2.4.2. Crítica de las aplicaciones educativas para Second Life y OpenSim

Los sistemas de educación virtual antes mencionados han ayudado a entender que este medio es útil para una enseñanza a distancia, aunque muchos de ellos se limitan a llevarlo a cabo como si fuese una clase magistral, aprovechando la facilidad de colaboración que brindan las plataformas, o presentando información correspondiente a la asignatura por medio de paneles o vídeos en un entorno similar al de la vida real.

Ninguno de éstos permite una interacción avanzada con los objetos 3D, a excepción de SWIFT, como manipular dichos objetos para cambiar sus estados, como encender una lámpara o llenar un vaso de agua, y así cumplir un objetivo.

El único estudio que propone algo cercano a una tutoría es SWIFT, pero restringe las opciones que puede seleccionar el alumno a sólo las correctas, es decir, si el estudiante se equivoca, le vuelve a presentar la pregunta hasta que seleccione la opción adecuada.

El presente trabajo, a diferencia de los estudios antes mencionados, permite realizar una práctica de biotecnología siguiendo el proceso, definido en el siguiente capítulo, a través del manejo y cambio de estado de varios objetos 3D. Estos objetos son coordinados por una tutoría que supervisa la correcta realización de dicha práctica y que permite que el estudiante cometa errores que muestran un resultados diferentes, y así pueda aprender mejor.

Capítulo 3

Planteamiento del Problema

3.1. Conceptos básicos de Biotecnología

La biotecnología es la ciencia que estudia el uso de procesos biológicos con fines industriales, su estudio implica el uso de técnicas de genética, biología molecular y bioquímica. Las principales áreas de aplicación biotecnológica son: médica, industrial y agro-forestal.

3.1.1. Manipulación genética

Las principales aproximaciones técnicas utilizadas para el estudio de la función de un gen consisten en la denominada “genética directa”, que consiste en producir un incremento en la expresión del gen de interés, y la “genética inversa”, generando la anulación de la expresión del gen bajo estudio.

3.1.2. Clonado de genes

Vector de clonación. Es una molécula de ADN que contiene información genética y que sirve de vehículo para la transferencia de esta información genética a un organismo hospedador. Los principales tipos de vectores de clonación utilizados son: plásmido, fago(*Phage*), cósmido, BAC (*Bacterial Artificial Chromosomes*), YAC (*Yeast Artificial Chromosomes*).

Gen marcador de resistencia a antibióticos. Aquél que concede resistencia a un antibiótico, acompaña al gen de interés en el vector de clonación. Permite la selección del organismo que lo contiene cuando éste crece en un medio en presencia de dicho antibiótico.

3.1.3. PCR (Reacción en Cadena de la Polimerasa)

La PCR, es una reacción enzimática que permite la obtención de ADN del gen de interés en alta concentración para su posterior estudio mediante una reacción específica de amplificación de dicho gen de interés. Está catalizada por la enzima Taq polimerasa, cuya temperatura óptima de actuación es de 72°C.

3.1.4. Transformación de plantas

Método mediante el cual se modifica la información genética (genoma) contenido en una célula vegetal.

Agrobacterium tumefaciens. Bacteria capaz de inducir la producción de tumores en plantas (*crown gall*). Éste contiene un plásmido (Ti o *tumor induction*) capaz de insertarse en el genoma de la planta e inducirle un tumor. Esta bacteria ha sido modificada genéticamente eliminando la información que causa el tumor en la planta, pero manteniendo su capacidad de insertarse en el genoma, lo que la hace una herramienta fundamental en biotecnología para la obtención de organismos modificados genéticamente.

Medio de cultivo. Medio de crecimiento de plantas en condiciones estériles (axénicas), es decir, en ausencia de ningún otro microorganismo. Suele prepararse en forma de gel o en solución, y contiene todos aquellos nutrientes esenciales para el desarrollo de la especie vegetal a cultivar. La composición específica adecuada para la especie que se va a cultivar en esta práctica es la siguiente:

- *Murashige and Skooh*. Mezcla comercial específica con vitaminas, minerales y otros nutrientes.
- Sacarosa. Azúcar (disacárido) utilizado en el medio como fuente principal de carbono para el metabolismo de la planta.
- Agar. Le sirve para dar consistencia gelatinosa al medio y permitir el anclaje de las raíces.
- Ácido Indolacético (IAA, aux). Es una hormona que induce la formación de raíces en el brote acelerando su crecimiento.

3.1.5. Cultivo *in vitro* de plantas

Micropropagación. Es una técnica de regeneración del material vegetal. En esta práctica se lleva a cabo a partir de brotes de un árbol de chopo de tres meses,

crecido mediante cultivo *in vitro*¹, para su trabajo en laboratorio. Se generan nuevos individuos en botes con un medio de crecimiento apropiado dentro de un fitotrón (cámara visitable de crecimiento de plantas), en condiciones de luz, humedad y fotoperiodo² controladas. El árbol crece durante varias semanas (hasta tres o cuatro) en estas condiciones. La técnica de micropropagación permite la obtención de nuevos individuos con el mismo contenido genético [Renneberg, 2008].

3.2. ¿Qué es un laboratorio de biotecnología?

Es aquél habilitado para llevar a cabo todos los experimentos relacionados con biotecnología, el cual debe disponer tanto de la maquinaria necesaria, como del material biológico y bioquímico.

El lugar que alberga un laboratorio de biotecnología normalmente dispone de varias salas enumeradas a continuación:

- Sala de reuniones.
- Biblioteca.
- Sala de seminarios.
- Sala de computadores.
- Sala de crecimiento.
- Laboratorios.

La sala de computadores, véase la figura 3.1, dispone del software necesario para facilitar los cálculos, así como permite realizar consultas en portales web especializados. La sala de crecimiento, véase la figura 3.2, es en la que se colocan los tubos de ensayo o botes con plantas para su crecimiento, ésta tiene un entorno favorable para el crecimiento de las plantas, es decir, tiene la luz y la temperatura indicadas para que crezcan exitosamente.

Los laboratorios se dividen por nivel de seguridad desde P1 hasta P4, donde los de nivel P1, como el de la figura 3.3, son los que necesitan menos protecciones de seguridad, como simplemente unos guantes de látex, mientras que en los de nivel P4 pueden existir patógenos en el aire y es necesario utilizar trajes protectores.

Algunos de los instrumentos y maquinarias utilizadas en esta práctica son las siguientes:

- Poyata. Es la mesa de trabajo
- Agitador. Sirve para mezclar líquidos y sólidos.

¹Cultivo de plantas en un envase de vidrio

²Tiempo de exposición a la luz



Figura 3.1: Sala de computadores en el Centro de Biotecnología y Genómica de Plantas de la UPM



Figura 3.2: Fitotrón en el Centro de Biotecnología y Genómica de Plantas de la UPM

- pH-metro. Mide la acidez o basicidad de una solución acuosa.
- Vaso de precipitados. Envase de vidrio o plástico milimetrado.
- Botella para autoclavar. Botella de vidrio especial que permite su esterilización.
- Autoclave. Permite la eliminación de microorganismos del material que se esteriliza en su interior.



Figura 3.3: Laboratorio en el Centro de Biotecnología y Genómica de Plantas de la UPM

- Cabina de flujo laminar. Es un receptáculo que permite trabajar en el laboratorio en un entorno estéril, ya que se filtra el aire que contiene, lo que es necesario para trabajar con cultivos de células o tejidos.
- Fitotrón. Cámara visitable de crecimiento de plantas, en condiciones de luz, humedad y fotoperiodo controladas, esto minimiza el impacto de las variaciones ambientales en el crecimiento y reduce las contaminaciones.
- Esterilizador de bolas. Permite la esterilización de instrumentación dentro de la cabina de flujo laminar.
- Micropipeta. Aparato que permite la toma y dosificación de volúmenes en cantidades muy pequeñas en laboratorio (a concentraciones micromolares).
- Tubo *ependorf*. Tubo de plástico autoclavable para la mezcla de reactivos en biología molecular utilizando volúmenes de hasta 2 ml.
- Placa petri. Recipiente de vidrio o plástico utilizado para el cultivo de plantas y otros organismos.
- *Falcon*. Tubo de plástico autoclavable para la mezcla de reactivos en biología molecular utilizando volúmenes de hasta 50 ml.
- Máquina de PCR. Realiza la reacción en cadena de polimerasa, que es un método para amplificar in vitro un segmento específico de ADN.
- Electroporador. Mediante la diferencia de potencial de su fuente eléctrica, se abren los poros en las paredes bacterianas para permitir el paso del ADN plasmídico al interior de éstas.

3.3. Ejercicio práctico a modelar

3.3.1. Objetivos de Aprendizaje

A lo largo de la práctica el alumno tiene acceso a material didáctico específico elaborado por el profesorado del departamento de Biotecnología de la UPM. A partir de ese material, y de la elaboración de la práctica, el alumno cumple varios objetivos de aprendizaje, relacionados con técnicas básicas y fundamentos de trabajo en un Laboratorio de Biotecnología vegetal, incluyendo:

- Cultivo en condiciones in vitro, micropropagación de material vegetal y trabajo en condiciones de esterilidad, precauciones a tomar para evitar contaminaciones y utilización del material de laboratorio apropiado.
- Manejo de medidor de pH, balanzas de laboratorio y agitadores.
- Manejo de cabinas de flujo laminar.
- Utilización de autoclave.
- Preparación de medios de cultivo.
- Manejo de micropipetas de laboratorio.
- Fundamentos y manejo de la técnica de PCR.
- Fundamentos y técnicas para el clonaje de genes: ligación y clonación.
- Manejo de fitotrones.
- Técnica de transformación bacteriana mediante electroporación.
- Técnica de transformación vegetal.
- Técnicas de inoculación in vitro de especies vegetales.
- Análisis fenotípico de síntomas.

La práctica concreta que se ha desarrollado permite al alumno reforzar estos conocimientos adquiridos con un ejemplo experimental en el que se estudia un gen que codifica para una proteína que es un antibiótico vegetal, que concede resistencia a los árboles frente a enfermedades que suelen afectar a éstos, especialmente las producidas por hongos.

Dicha práctica está dividida en varias fases, para que el estudiante en cada una de éstas pueda adquirir conocimientos y destrezas de biotecnología que le ayudaran para realizar la siguiente parte de la práctica.

3.3.2. Protocolo de la Práctica

Es la receta que debe seguirse paso a paso y de forma estricta, para obtener un resultado que se pueda reproducir o repetir. Es un elemento fundamental en el cumplimiento del método científico, pues al ejecutarlo y posteriormente repetirlo se puede confirmar o desestimar la hipótesis planteada.

Las fases a seguir del protocolo de la presente práctica son:

1. Micropropagación del material vegetal. A partir de una planta no modificada genéticamente se obtiene varias.
2. PCR. En este paso se amplifica un gen de interés de un ADN molde de Chopo.
3. Ligación. Se introduce el gen amplificado por la PCR en el plásmido.
4. Transformación bacteriana. Se inserta el plásmido en las bacterias.
5. Transformación vegetal. En esta fase se obtiene el árbol modificado genéticamente, mediante la infección de éste con las bacterias al introducir el gen amplificado.
6. Inoculación de tejido vegetal. Se inocula con hongos la planta modificada y otra silvestre y se comparan los efectos producidos. Así se comprueba si la planta ha sido modificada genéticamente con éxito.

Micropropagación del material vegetal

1. Preparación del medio de cultivo o crecimiento.
 - a) Tomar un vaso de precipitados y llenarlo de agua destilada en el grifo.
 - b) Ir a la mesa de trabajo donde se encuentra el agitador, colocar el vaso sobre el agitador y añadir los siguientes ingredientes por este orden:
 - 1X Murashige Skooh -S1b.
 - 2 % Sacarosa.
 - c) Ajustar el pH-metro a 5,8 colocando el vaso en el medidor de pH-metro.
 - d) Añadir 0,8 % agar, pH de 5,8.
2. Pasar el medio a una botella (ponerle una cinta de autoclave en la tapadera), y llevarla al autoclave.
3. Recoger una bandeja con botes estériles y la botella con medio del autoclave, y llevarlos a la cabina de flujo laminar.
4. Encender la cabina de flujo laminar, y esterilizar las pinzas y tijeras poniéndolas en el esterilizador de bolas de vidrio. La cabina se debe encender con

CAPÍTULO 3. PLANTEAMIENTO DEL PROBLEMA

antelación, para que el aire en ella se encuentre esterilizado y así trabajar en un entorno estéril.

5. Una vez que el medio está atemperado añadir IAA, (0,5 mg/L), no se debe esperar demasiado pues si el medio se endurece será imposible añadirlo.
6. Repartir 2cm del medio en cada bote estéril y dejar solidificar (cambia el color de amarillento a grisáceo).
7. Sacar el árbol con las pinzas y colocarlo sobre una placa petri de cristal, cortarlo en brotes con al menos una hoja cada uno y transferir cada brote a un tubo, clavándolo en el medio.
8. Llevar los nuevos brotes al fitotrón y dejarlos durante seis semanas, allí estarán a 25°C con un fotoperiodo de 16 horas de luz y ocho horas de oscuridad.

PCR

1. Encender la máquina de PCR, su termobloque tarda en calentarse.
2. Recoger el soporte con hielo y tomar del frigorífico a -20°C los siguientes productos:
 - ADN, molde a partir del cual se amplificará el gen (100ng/ul).
 - Oligonucleótidos (primer) derecho e izquierdo 100uM (*forward and reverse*).
 - Tampón de la enzima polimerasa.
 - Cloruro de Magnesio 25mM.
 - Mezcla estándar de dNTP 1mM (dinucleótidos trifosfato).
 - ADN polimerasa (5u/ul).
 - Agua destilada.
3. Ir a la poyata y en un tubo *eppendorf* de PCR mezclar los componentes utilizando una pipeta.
4. Introducir la mezcla en la PCR y pulsa el botón para comenzar, cuando acabe la PCR guardar el fragmento amplificado en un frigorífico a -20°C.

Ligación

1. Encender el termobloque a 16°C.
2. Recoger el soporte con hielo y tomar del frigorífico a -20°C los siguientes productos:

- ADN plasmídico (incluye un gen de resistencia a antibiótico, una secuencia promotora que regula la expresión del gen y una zona para la inserción del gen).
 - ADN amplificado por PCR (obtenido en la fase anterior).
 - Tampón de la enzima ligasa.
 - Enzima ligasa.
 - Agua destilada.
3. Ir a la poyata y en un tubo *eppendorf* mezclar todos los componentes utilizando una pipeta.
 4. Introducir el tubo con la mezcla en el termobloque y dejarlo toda la noche (*overnight*), es decir, de ocho a diez horas a 16°C.
 5. Guardar el plásmido en un frigorífico a -20°C.

Transformación bacteriana

1. Recoger el soporte con hielo y sacar un tubo de bacterias de un frigorífico a -80°C y dejarlo en el hielo, las bacterias deben descongelarse lentamente para disminuir la tasa de mortalidad.
2. Tomar del frigorífico a -20°C el ADN plasmídico que contiene el gen, y dejarlo en el hielo.
3. Ir al aparato de electroporación que está dentro de la cabina, depositar en ella una mezcla del plásmido y de la bacteria que se acaba de descongelar. El choque eléctrico abre las paredes bacterianas y el plásmido entra en la bacteria.
4. Colocar el tubo en el horno a 37°C durante una hora, para que las bacterias se recuperen.
5. Pasar las bacterias del tubo a una placa petri con un medio de cultivo en presencia de antibiótico. Sólo crecerán las bacterias que contienen el gen con el plásmido que le confiere resistencia a dicho antibiótico.
6. De la placa tomar una colonia que crece en un medio líquido en el horno a 37°C y pasarlo a un tubo *falcon*, el cual va a contener la bacteria lista para transformar la planta.

Transformación vegetal

1. Recoger una planta de chopo del fitotrón (obtenido en la fase de micropropagación) y llevarla a la cabina de flujo laminar.

2. Sacar del bote y cortarlo para obtener fragmentos de hojas.
3. Pasar los fragmentos al *falcon*, obtenido de la ligación, para infectarlos con la bacteria. Con esto, las hojas contendrán el gen en su genoma.
4. Pasar los fragmentos a una placa de inducción de callo.
5. Llevar las placas al fitotrón, hasta que se formen los callos en unas seis semanas.
6. Transferir el callo diferenciado de la placa a un bote estéril con un medio de inducción de brotes.
7. Llevar los botes al fitotrón, hasta que se formen los tallos, en unas dos semanas.
8. Transferir cada tallo que ha crecido sobre el callo a un bote grande con un medio de crecimiento y llevarlo al fitotrón hasta que crezca el árbol modificado (unas tres semanas).

Inoculación de tejido vegetal

1. Ir al fitotrón y recoger la bandeja con los árboles modificados y una bandeja de árboles silvestres.
2. Ir a la cabina de inoculación, allí se encontrará un bote de *spray* con el inóculo de hongos preparado e infectar los árboles con el hongo.
3. Llevar los árboles al fitotrón y vigilar hasta que aparezcan los síntomas de enfermedad, si es que éstos aparecen.
4. Comparar las plantas silvestres y las transformadas con el gen, y anotar los resultados para poder obtener conclusiones. Además de estimar en función de su fenotipo, si las plantas son resistentes ó susceptibles.

3.4. Descripción del laboratorio virtual

En la actualidad, realizando el laboratorio de forma manual, la cantidad de estudiantes que lo pueden ejecutar son pocos debido a algunas limitaciones. Por lo tanto mediante el laboratorio virtual presentado se aspira a conseguir los objetivos de aprendizaje con una cantidad mayor de estudiantes.

3.4.1. Justificación de la solución virtual

La necesidad de desarrollar un entorno virtual para este laboratorio se basa en tres motivos fundamentales: el costo, el tiempo que demora, y evitar los riesgos a los que pueden estar expuestos los estudiantes en un laboratorio real.

Para realizar esta práctica en un laboratorio de biotecnología se deben comprar productos químicos, plantas y material de laboratorio desechable que son muy costosos y delicados, cuyos importes fluctúan entre los 10.000 € a 24.000 €. Por esto en las prácticas presenciales no se permite a los alumnos utilizarlos.

Si se desea hacer el mismo experimento al mismo tiempo con varios alumnos se necesita adquirir varios ejemplares de este material, lo cual multiplicaría los costes por el número de estudiantes. El laboratorio al ser virtual permite que todos los alumnos tengan acceso al equivalente virtual del material de laboratorio necesario.

Al no tener que comprar el material de laboratorio necesario para realizar la práctica, evidentemente los costos se reducen de manera abismal, y los conceptos aprendidos por los estudiantes son prácticamente los mismos.

Además, al manejar químicos o maquinarias, los estudiantes que no tengan experiencia pueden tener accidentes y salir malheridos, lo que no puede ocurrir en un entorno virtual.

El otro motivo que justifica el uso de este tipo de laboratorio es reducir el tiempo de duración del experimento, pues al ser un entorno virtual se puede acelerar, tanto como se desee, el crecimiento de la planta que normalmente dura alrededor de tres meses; así el tiempo total que requiere esta práctica puede ser de dos años, mientras que en un entorno virtual se puede realizar toda la práctica en menos de una hora.

Al transformar la práctica real en un sistema de software, se puede tener un mayor control sobre las acciones que realiza el estudiante, así como también es posible obtener datos de su comportamiento o creatividad, aspectos que pueden pasar desapercibidos de forma presencial.

La realización de la práctica virtual pretende reforzar los conocimientos adquiridos en clase por el estudiante de manera complementaria a las prácticas presenciales, y no pretende de ningún modo sustituirlas.

3.4.2. Requisitos

El entorno virtual debe cumplir las condiciones que se citan a continuación:

Generales:

- De acuerdo a lo investigado en el apartado 2.2, se debe utilizar la plataforma OpenSimulator como herramienta para la creación del entorno.
- El laboratorio debe ser representado por un modelo tridimensional que reproduzca, lo más parecido a la realidad posible, el material de laboratorio, la maquinaria, los instrumentos y las salas que se utilizan para realizar el experimento.

- Este entorno debe permitir la comunicación síncrona con los demás participantes, ya sean estos estudiantes o profesores, para favorecer el trabajo en equipo.
- El entorno debe permitir realizar el experimento a varios estudiantes.
- Se debe tener un libro de fundamentos y otro de protocolos, que al ser tocados muestran los fundamentos y protocolos de la práctica mediante un proyector o pizarra.
- El crecimiento de las plantas se debe acelerar, de forma que al salir y volver a entrar a la sala de crecimiento se debe avanzar una nueva fase de crecimiento. El crecimiento de la planta se dividirá en 4 fases.

Validación de acciones del usuario

- Se deben registrar las acciones que realicen cada uno de los alumnos en el entorno virtual.
- Se deben validar las acciones que realiza el avatar, es decir, cuando el avatar intente realizar una acción sobre algún objeto se debe validar si la acción se puede hacer o no.
- Se debe poder proporcionar ayuda al estudiante sobre la forma de interactuar con el entorno para realizar la siguiente acción del proceso.
- Para controlar la correcta realización de la práctica es necesario que cuando se desee ejecutar una acción se valide que la anterior haya sido realizada, creando así una dependencia entre ellas.
- Una acción puede requerir que se hayan realizado una o más acciones anteriormente. Si son varias, se podría requerir que estas acciones se hubieran realizado en un cierto orden.
- Si al realizar una acción no se han realizado las acciones que deberían haberse realizado anteriormente para que el proceso sea correcto, se pueden dar dos tipos de errores: el primero, más restrictivo, que no permite que la acción actual se ejecute; y el otro que sí lo permite, pero al finalizar una fase provoca un resultado indeseado. En este último caso, además, se mostrarán al estudiante todos los errores cometidos y se le dará la oportunidad de reiniciar la fase.
 - Los errores que permiten proseguir con la práctica son los siguientes:
 - Olvidar tomar una bata.
 - Olvidar tomar guantes de látex.
 - Agregar Murashige&Skoog a la mezcla.
 - Agregar sacarosa a la mezcla.

- Ajustar el pH de la mezcla.
- Agregar agarosa a la mezcla.
- Encender la cabina de flujo.
- No esperar el tiempo necesario para que se esterilice el aire de la cabina.
- No esperar el tiempo necesario para que se atempere la mezcla en la botella después de soltarla en la cabina.
- Estas acciones deben ser fácilmente configurables. Para esto se utilizará un archivo de texto donde se especifican. Además, se detalla por cada una de éstas las acciones anteriores de las que dependen y los mensajes necesarios que se deben presentar.
- El presentar o no uno u otro mensaje al estudiante, ya sean de ayuda, de error o de acción realizada debe ser fácilmente configurable.
- El alumno podrá borrar las acciones de la última fase registradas para volverla a hacer.

Interacción del usuario con los objetos:

- La utilización de instrumental como vasos, botes, placas, entre otros, debe ser individual por cada avatar, es decir, cada avatar debe obtener uno de estos y utilizarlo. Para ello cada objeto debe permanecer en el estado del proceso en el que lo dejó su avatar.
- Mientras un avatar esté utilizando algún objeto como la mesa de trabajo, el agitador, el fregadero, el pH-metro, el autoclave o la cabina de flujo, debe restringirse el uso de dicho objeto para que otros avatares no lo puedan utilizar.
- Los objetos del entorno deben tener permisos para que solo un usuario administrador pueda modificarlos. De esta manera, se pretende evitar que un alumno pueda deliberada o accidentalmente dañar el laboratorio virtual.
- Cuando un objeto ha dejado de utilizarse, debe eliminarse del entorno automáticamente. Esto no debe ocurrir con objetos grandes como vitrinas, sino con objetos como el vaso de precipitados, la botella para autoclavar, el bote para crecimiento o la bandeja con planta.
- Para interactuar con cualquier objeto presente en el entorno, el objeto debe ser tocado por el avatar.
- Para interactuar con cualquier objeto adjunto a la mano del avatar, es necesario soltarlo primero.

CAPÍTULO 3. PLANTEAMIENTO DEL PROBLEMA

- Para pasar alguna substancia o material vegetal de un contenedor a otro, se debe tocar el recipiente de destino.
- Cuando el avatar adjunte un objeto a su mano, ésta debe tomar la postura correspondiente para sostener dicho objeto, si se trata de un objeto pequeño, o las dos manos si es grande.
- Cuando se necesite obtener un objeto, como un vaso de precipitados o una botella para autoclavar desde la vitrina, se debe escoger de una lista presentada después de tocar la puerta. Lo mismo debe hacerse para agregar algún producto químico ya sea del frigorífico o de la vitrina de químicos.
- La botella para autoclavar solo podrá ser soltada por un alumno cerca del agitador o la cabina de flujo bloqueada para el mismo alumno. La bandeja de botes y el bote con planta lo podrán hacer solo cerca de la cabina.

Capítulo 4

Solución adoptada

4.1. Introducción

El objetivo de este capítulo es definir la arquitectura y el diseño del entorno virtual que permita llevar a cabo la práctica descrita en la sección 3.3.

Para este trabajo solamente se diseña la primera fase del proceso del laboratorio descrito en el protocolo de la sección 3.3. Ésta es la fase de micropropagación del material vegetal (3.3.2), en la que partiendo de una planta inicial -y con la ayuda de un medio de enraizamiento adecuado- se crean varias plantas de similares características a la inicial.

Los modelos o tablas realizados para definir el diseño son: (1) modelo del protocolo, que muestra el flujo correcto de las acciones que se deben realizar en todo el proceso; (2) modelo de dominio, en el cual se presentan las relaciones entre todos los objetos creados; (3) diseño de los objetos activos, donde se especifican los mensajes y los eventos recibidos, las actividades realizadas y los mensajes enviados como respuesta a estos mensajes y eventos; y (4) diseño de la interacción entre objetos, representado con diagramas de secuencia parecidos a los utilizados en UML.

4.2. Descripción de la Arquitectura

Para el desarrollo de este proyecto se utilizó la plataforma para mundos virtuales OpenSim, cuya arquitectura se describe en el apartado 2.2.3, instalado sobre un sistema operativo Windows 7 y con una base de datos MySQL 5.5¹.

Todo el proceso definido en el apartado 4.4, se encuentra dividido en fases, que son agrupaciones de acciones según su afinidad. El estudiante puede eliminar la última

¹<http://www.mysql.com/downloads/mysql/>

fase si ha cometido algún error y necesita realizar ésta de nuevo.

Los objetos 3D diseñados intervienen en dicho proceso, ya sea por interacción con el avatar por medio de eventos o por envío y recepción de mensajes entre ellos, detallados en el apartado 4.7. Otros objetos, como mobiliario o maquinaria, no se destruyen, pero mientras son utilizados por un estudiante se bloquean, y presentan un texto con su nombre, para que otros no los puedan utilizar ni modificar lo que se está realizando en él. Estos objetos además tienen los permisos necesarios de modo que solamente el usuario administrador puede alterar su forma o cambiarlos de lugar.

Como se mencionó, el avatar interactúa con los objetos por medio de eventos como tocar, sentarse, adjuntar el objeto a la mano y soltarlo de la mano. Este evento es capturado por el objeto para ejecutar una acción determinada; por ejemplo, el avatar puede tocar una puerta para que ésta se abra, o puede soltar un vaso que sostiene en la mano para que se posicione en una mesa.

Cuando se necesite obtener un vaso de precipitados o una botella, el avatar debe acercarse a una vitrina y tocar la puerta, ésta mostrará una lista donde se podrá seleccionar el material necesario. Después la vitrina entregará al inventario de dicho avatar el objeto seleccionado, haciéndolo propietario del mismo.

Para movilizar de un lugar a otro un objeto del tipo antes mencionado debe ser adjuntado a la mano del avatar y después se debe mover al avatar al lugar deseado, aquí se debe soltar el objeto para poder interactuar con él tocándolo o tocando otros objetos dependiendo de la acción del protocolo que se desee realizar. Según el proceso, cuando uno de estos objetos ha terminado su vida útil, dentro del mundo virtual, se destruye automáticamente, para así no dejar objetos “basura” dentro del entorno, y que posteriormente puedan estorbar a otros usuarios. De igual manera, estos objetos tendrán los permisos necesarios para que el usuario no pueda alterar su forma, pero que sí pueda moverlos de posición.

En algunas tareas del proceso intervienen dos o más objetos, para esto es necesario que éstos intercambien mensajes, con el fin de compartir valores de variables o comunicarse entre ellos. Estos mensajes deben contener el id del avatar que ejecuta la acción, para así poder filtrar los mensajes que envían los objetos de otros avatares. En muchas de estas acciones se mueven productos químicos o vegetales de un recipiente a otro, para realizarlas se debe tocar el recipiente de destino.

Otras tareas necesitan un tiempo mínimo para ejecutarse por completo o un tiempo máximo para que se ejecute la siguiente acción, por lo cual se utiliza un temporizador por cada una de éstas por el tiempo necesario.

Para validar la correcta realización del protocolo, definido en la sección 4.4, se ha implementado un objeto llamado “tutor”, detallado en la sección 4.6.2, que valida la acción que se desea realizar. Este objeto verifica que el estudiante haya realizado ya las acciones anteriores a la actual que se consideran precondiciones de ésta, si no

las ha realizado todas, existen dos tipos de errores: el primero permite que la acción se ejecute, pero anota dicho error para posteriormente mostrárselo al estudiante; mientras que el otro no permite que la acción se realice hasta que el avatar haya realizado todas las acciones que se requieren como precondiciones de la actual.

Para que el tutor pueda validar las acciones, guarda cada una de éstas en un archivo por avatar. Además, presenta ayuda al alumno, para esto presenta un mensaje que muestra cuál es la siguiente acción que debe realizar y cómo se debe llevar a cabo.

4.3. Modelos 3D realizados

Los modelos 3D que se utilizan en la fase del laboratorio descrita en este apartado fueron construidos por el autor, excepto algunos objetos que forman parte del decorado como el edificio, muebles del vestíbulo, mesas del patio, flores y cerramiento de la laguna que fueron descargados de la página web *Opensim Creations*². Otros objetos que se van a utilizar en las demás fases del laboratorio -que actualmente se encuentran de decorado- como micropipetas con puntas, horno, PCR, microfuga, tubos eppendorf, bandeja con hielo y microscopio confocal fueron construidas por la empresa Novatierra³.

4.3.1. Edificio

Es el lugar en el que se encuentran todos los objetos para realizar la práctica, está dividido en varias zonas: vestíbulo, patio, sala principal del laboratorio, sala de cabinas de flujo, sala de autoclavado y sala de crecimiento o fitotrón. Éste se muestra en la figura 4.1.

Vestíbulo (figura 4.2). Es la primera habitación del edificio, y se encuentra diseñado para la interacción entre los estudiantes.

Patio (figura 4.3). Zona externa al edificio también diseñada para la interacción entre los estudiantes.

Sala principal del laboratorio (figura 4.4). En ésta los estudiantes disponen de su espacio de trabajo, así como también de maquinaria, vitrinas con instrumental plástico y de vidrio, además de vitrinas y frigoríficos con productos químicos.

²<http://opensim-creations.com/>

³<http://www.novatierra.com>



Figura 4.1: Edificio

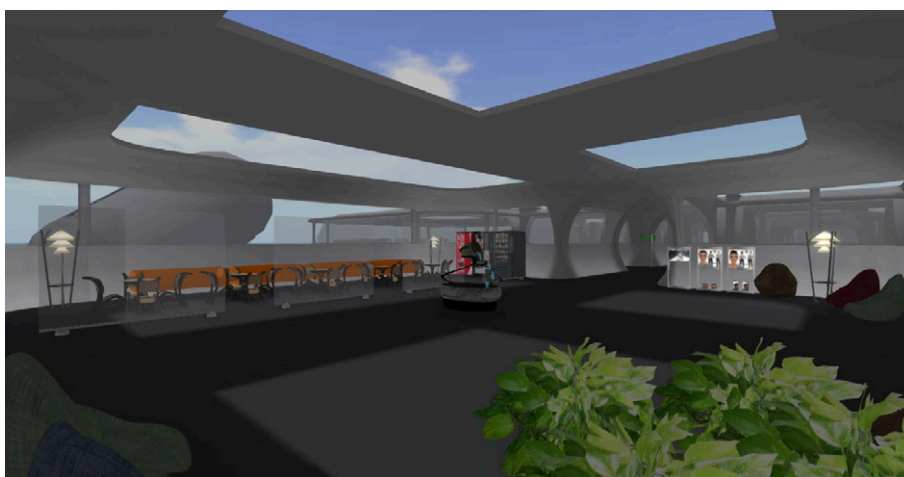


Figura 4.2: Vestíbulo



Figura 4.3: Patio



Figura 4.4: Sala principal del laboratorio

Sala de cabinas de flujo laminar (figura 4.5). Habitación en la que se encuentran las cabinas de flujo para trabajar en un entorno estéril.



Figura 4.5: Sala de cabinas de flujo laminar

Sala de autoclavado (figura 4.6). Sala en la que hay autoclaves para la esterilización y limpieza del material, además de instrumental autoclavado como bandejas de botes para la micropropagación de material vegetal.

Fitotrón o sala de crecimiento (figura 4.7). Habitación para el crecimiento de plantas, que tiene condiciones de luz, humedad y fotoperíodo controladas. En ésta los estudiantes pueden obtener botes con plantas crecidas o colocar sus bandejas para que crezcan sus plantas.



Figura 4.6: Sala de autoclavado

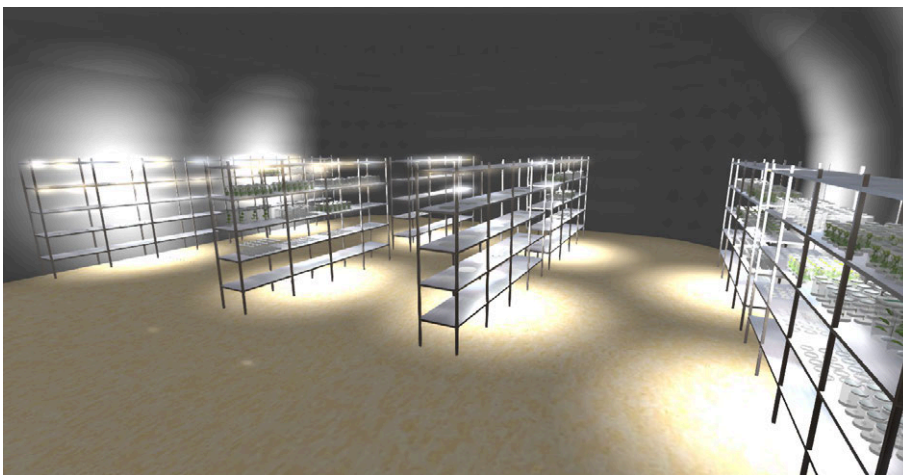


Figura 4.7: Fitotrón o sala de crecimiento

4.3.2. Objetos

Los siguientes objetos se encuentran colocados en las salas del edificio de biotecnología antes mencionados.

Seleccionador de práctica (figura 4.8). Este objeto se encuentra en el vestíbulo del edificio y presenta un diálogo en el cual el estudiante puede seleccionar la práctica, con la vista puesta a que en el futuro se agreguen otras. Al seleccionarla se crea el cuaderno de protocolo del alumno en el tutor explicado en el apartado 4.6.2.

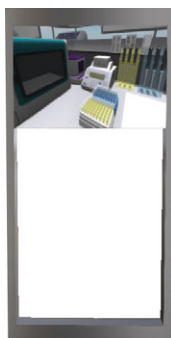


Figura 4.8: Seleccionador de práctica

Poyata (figura 4.9). Es donde los estudiantes comienzan a realizar la práctica de laboratorio. Aquí pueden estudiar los fundamentos, el protocolo de la práctica y los conceptos básicos. Ésta contiene varios objetos, un escritorio, una pizarra, una silla, dos libros, un PCR, un horno, una microfuga y un botón para eliminar la última fase realizada del proceso.

Pizarra (figura 4.10). en ésta se muestra una diapositiva de bienvenida, además de los fundamentos, el protocolo de la práctica, y los términos de Wikipedia según el libro que se haya tocado. Estas diapositivas tendrán botones para navegar entre éstas, es decir, un botón para avanzar a la siguiente diapositiva y otro para retornar ala anterior.

Silla (figura 4.11). el alumno se puede sentar en ella solamente tocándola o mediante el sub-menú “Sentarse Aquí” del menú contextual, después de sentarse la cámara se colocará de forma que se puedan ver mejor la pizarra, los libros y el instrumental de laboratorio que se encuentra en la mesa.

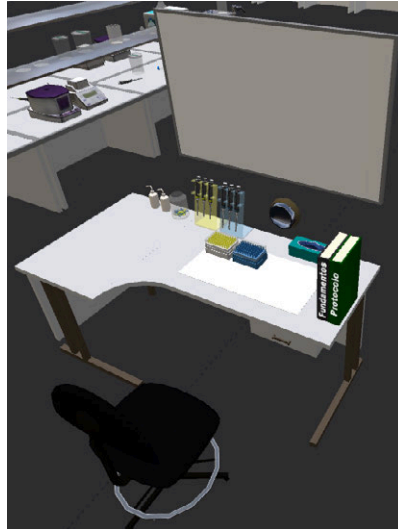


Figura 4.9: Poyata

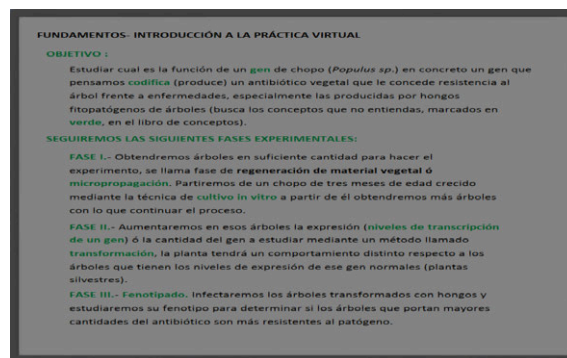


Figura 4.10: Pizarra

Libros (figura 4.12). el primer libro es el de “Fundamentos”, al tocarlo se muestra en la pizarra los fundamentos del laboratorio; el segundo libro es el de “Protocolo”, el cual muestra el protocolo de la práctica. Tanto los fundamentos como el protocolo poseen vínculos a definiciones de algunos términos.



Figura 4.11: Silla



Figura 4.12: Libros

Caja de guantes de latex (figura 4.13). Contiene guantes de color magenta, que se entrega al avatar una vez que éste la ha tocado. Estos guantes se entregan al inventario del avatar, de donde se los puede poner en las manos.

Percha con batas (figura 4.14). Tiene batas blancas de hombre y de mujer, para usarlas en todo el desarrollo del laboratorio. Para obtener una, el estudiante debe tocar la bata que desee, aceptar el mensaje de permiso, y después ponérsela presionando dos veces sobre ella desde su inventario.

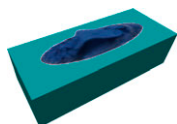


Figura 4.13: Caja de guantes de latex



Figura 4.14: Pecha con batas

Lavamanos (figura 4.15). Al ser tocado por el avatar crea una animación de agua saliendo por el grifo.



Figura 4.15: Lavamanos

Vitrina de instrumentos (figura 4.16). Al tocar la puerta de la vitrina el alumno puede obtener diferentes instrumentos como el *Vaso de Precipitados*, *Bandeja de botes*, *Placa*, *Botes de crecimiento* o *Falcon*, los cuales se adjuntan a la mano del avatar después de aceptar el mensaje de permiso.

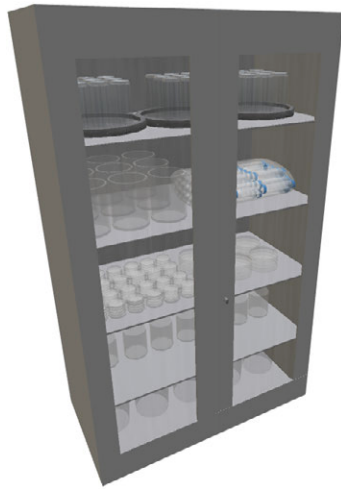


Figura 4.16: Vitrina de instrumentos

Vaso de precipitados (figura 4.17). en este vaso se crea el medio de crecimiento de las plantas. Éste se lo debe adjuntar a la mano para posteriormente soltarlo en el fregadero o posteriormente en el agitador.

Bandeja de botes (figura 4.18). en los botes de esta bandeja crecen los árboles que se han micropropagado. Para interactuar con ella, primero se la debe adjuntar a la mano, después se la suelta en la cabina de flujo o en el fitotrón, una vez que se ha colocado en la cabina se la puede tocar.



Figura 4.17: Vaso de precipitados

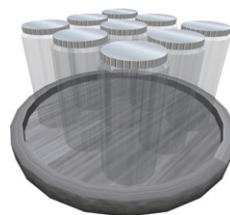


Figura 4.18: Bandeja de botes

Fregadero (figura 4.19). Al tocar la válvula el vaso que se encuentre en éste se llenará de agua destilada.

Agitador (figura 4.20). Para encenderlo el alumno debe tocarlo, esto hará que el agua dentro del vaso que se encuentre sobre su base gire en forma de remolino; para apagarlo también hay que tocarlo.



Figura 4.19: Fregadero

pH-metro (figura 4.21). Se encuentra junto al agitador, y cuando el alumno lo toca mide el pH de la mezcla del vaso que se encuentra sobre dicho agitador.

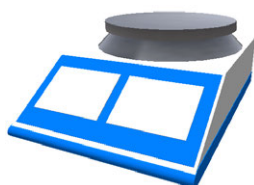


Figura 4.20: Agitador

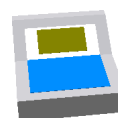


Figura 4.21: pH-metro

Frigorífico (figura 4.22). Contiene productos que necesitan ser refrigerados a -4°C (puerta superior) y -20°C (puerta inferior). Para agregar a la mezcla uno de éstos simplemente se debe tocar la puerta y seleccionarlo de la lista.

Vitrina de productos químicos (figura 4.23). Contiene productos químicos que no necesitan ser refrigerados, para agregar uno de éstos a la mezcla simplemente se debe tocar la puerta y seleccionarlo de la lista.

Botella para autoclavar (figura 4.24). En ésta se vierte la mezcla contenida en el vaso de precipitados para que pueda ser autoclavada.

Autoclave (figura 4.25). Esteriliza los recipientes que van a ser utilizados. Para ello, simplemente hay que solar el objeto cerca, para este caso la botella, después se la toca para encenderla y posteriormente se la vuelve a tocar para sacar el objeto que está dentro.



Figura 4.22: Frigorífico



Figura 4.23: Vitrina de productos



Figura 4.24: Botella para autoclavar

Cabina de flujo laminar (figura 4.26). Sirve para esterilizar el aire y de esta forma poder trabajar en un entorno limpio, para encenderla y apagarla hay que tocar los botones que se encuentran en la parte superior. Además, se debe tomar en cuenta que después de encenderla hay que esperar el tiempo necesario para que se esterilice el aire dentro de ésta, antes comenzar a trabajar en la misma.



Figura 4.25: Autoclave



Figura 4.26: Cabina de flujo laminar

Esterilizador (figura 4.27). Se encuentra en la cabina de flujo y sirve para esterilizar el instrumental que se utilizará para cortar los árboles. Para interactuar con él simplemente hay que tocarlo.



Figura 4.27: Esterilizador



Figura 4.28: Bote con planta

4.4. Modelo del Protocolo

En este apartado se presenta un modelo del protocolo presentado en la sección 3.3.2, los pasos o acciones definidos en este, son fundamentales para la realización del laboratorio y la obtención de un resultado satisfactorio. Estas acciones se las

agrupa en las dos tareas definidas en la figura 4.29. A su vez, como se indicará a continuación, cada una de estas tareas se va a descomponer en subtareas.

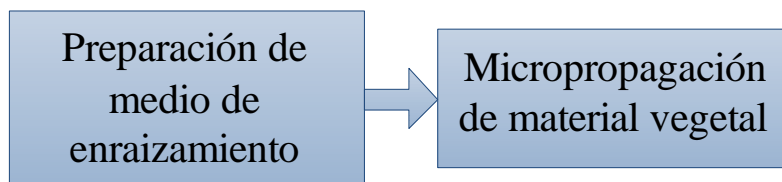


Figura 4.29: Macro procesos

Algunas de estas subtareas, detalladas en los siguientes gráficos como acciones de color verde, se definen como procedimientos o subtareas parametrizadas, debido a que su proceso es siempre el mismo pero aplicado a un objeto distinto X. Los procedimientos que existen para el presente protocolo se pueden observar en la figura 4.30.

Las flechas indican el orden en el que se deben ejecutar las acciones, mientras que si dos o más acciones están encerradas en una caja sin flechas que las conecten, significa que pueden realizarse en cualquier orden, siempre y cuando todas sean ejecutadas después de la acción o la subtaska anteriores y antes de las siguientes indicadas en la secuencia del protocolo.

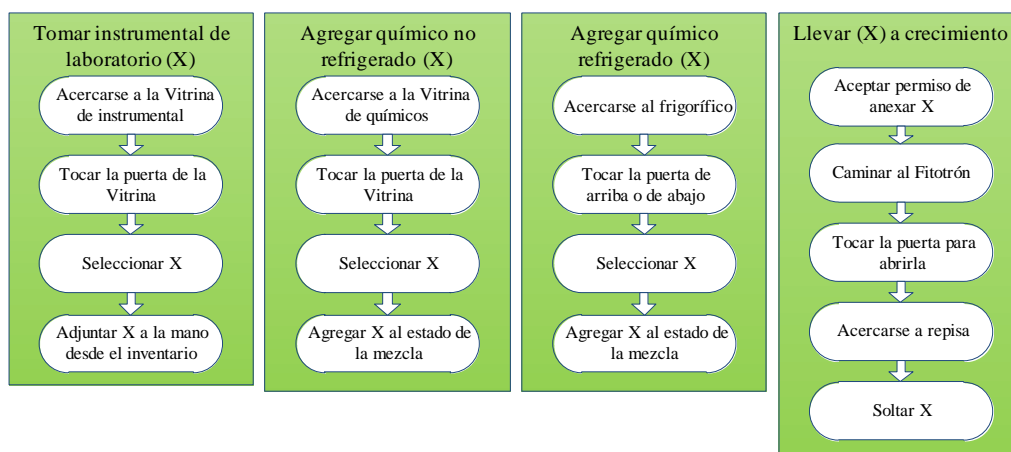


Figura 4.30: Procedimientos

La primera tarea llamada “Preparación de medio de enraizamiento” se subdivide en varias acciones definidas en la figura 4.31.

De igual manera, el segundo macro proceso llamado “Micropropagación de material vegetal” posee otras acciones que se muestran en la figura 4.32.

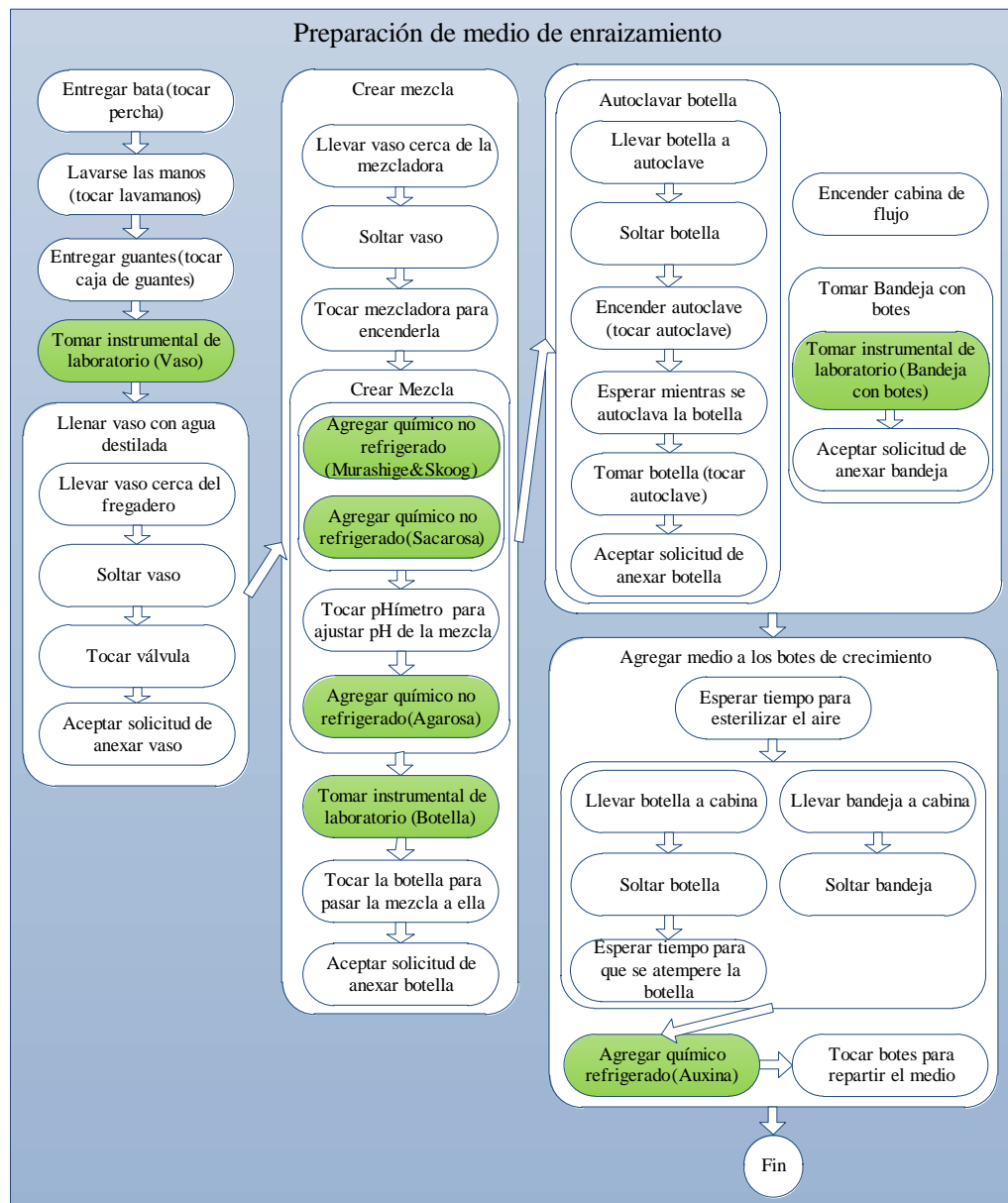


Figura 4.31: Preparación de medio de enraizamiento

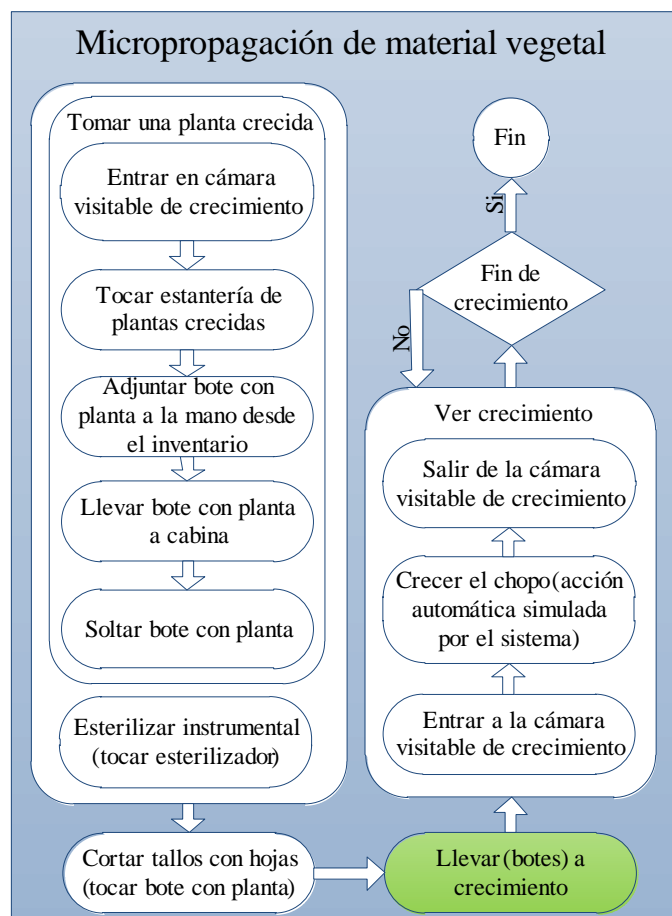


Figura 4.32: Micropropagación de material vegetal

4.6. Diseño de los Objetos Activos

4.6.1. Descripción de los Objetos Activos

Los objetos activos representan a objetos de la práctica con los cuales interacciona el avatar, para ejecutar alguna acción y así cumplir el protocolo definido en la sección 4.4.

En las siguientes tablas se detalla por cada objeto las actividades que se ejecutan, los eventos que pueden recibir por el avatar o por otros objetos, y los mensajes que envía o recibe. Por cada evento o mensaje recibido se especifica la actividad que se ejecuta en el objeto, y por cada actividad el mensaje que envía a otro objeto (si es necesario en el proceso).

Una actividad disparada es el proceso que se ejecuta dentro de un objeto para realizar una acción del protocolo de la práctica.

Seleccionador de práctica

Cuadro 4.1: Descripción del seleccionador de práctica

Mensaje recibido	Evento recibido	Actividad disparada	Mensaje enviado
	tocar	crearDiálogoPrácticas	
seleccionarPráctica1		crearProtocoloPráctica1	crearProtocoloPráctica1

Percha con batas

Cuadro 4.2: Descripción de la percha con batas

Mensaje recibido	Evento recibido	Actividad disparada	Mensaje enviado
	tocar	validarEntregarBata	validarEntregarBata
entregarBataVálido		entregarBata	

Silla de la Poyata

Cuadro 4.3: Descripción de la silla de la poyata

Mensaje recibido	Evento recibido	Actividad disparada	Mensaje enviado
	sentarse	cambiarCámara, validarBloquearPoyata	validarBloquearPoyata
bloquearPoyataVálido		bloquearPoyata, encenderPizarra	encenderPizarra, bloquearPoyata
	levantarse	apagarPoyata	apagarPoyata

Pizarra de la Poyata

Cuadro 4.4: Descripción de la pizarra de la poyata

Mensaje recibido	Actividad disparada
encenderPoyata	mostrarBienvenida
mostrarFundamentos	mostrarURLdeFundamentos
mostrarProtocolo	mostrarURLdeProtocolos
apagarPoyata	limpiarPizarra, apagarPoyata

Libro de Fundamentos

Cuadro 4.5: Descripción del libro de fundamentos

Mensaje recibido	Evento recibido	Actividad disparada	Mensaje enviado
bloquearPoyata	tocar	bloquearLibro validarLeerFundamentos	validarLeerFundamentos
leerFundamentosVálido		aumentarBrillo, mostrarFundamentos	mostrarFundamentos
mostrarProtocolo		quitarBrillo	
apagarPoyata		quitarBrillo	

Libro de Protocolo

Cuadro 4.6: Descripción del libro de protocolo

Mensaje recibido	Evento recibido	Actividad disparada	Mensaje enviado
bloquearPoyata	tocar	bloquearLibro validarLeerProtocolo	validarLeerProtocolo
leerProtocoloVálido		aumentarBrillo, mostrarProtocolo	mostrarProtocolo
mostrarFundamentos		quitarBrillo	
apagarPoyata		quitarBrillo	

Caja de Guantes

Cuadro 4.7: Descripción de al caja de guantes

Mensaje recibido	Evento recibido	Actividad disparada	Mensaje enviado
bloquearPoyata	tocar	bloquearCajaGuantes validarEntregarGuantes	validarEntregarGuantes
entregarGuantesVálido		entregarGuantes	

Lavamanos

Cuadro 4.8: Descripción del lavamanos

Mensaje recibido	Evento recibido	Actividad disparada	Mensaje enviado
	tocar	validarLavarseManos	validarLavarseManos
lavarseManosVálido		fluirAgua	

CAPÍTULO 4. SOLUCIÓN ADOPTADA

Vitrina de instrumentos

Cuadro 4.9: Descripción de la vitrina de instrumentos

Mensaje recibido	Evento recibido	Actividad disparada	Mensaje enviado
	tocar	abrirPuerta, crear-DiálogoInstrumental	
	seleccionarVaso	validarEntregarVaso	validarEntregarVaso
entregarVasoVálido		cerrarPuerta, entregarVaso	entregarVaso
	seleccionarBotella	validarEntregarBotella	validarEntregarBotella
entregarBotellaVálido		cerrarPuerta, entregarBotella	entregarBotella

Vitrina de productos

Cuadro 4.10: Descripción de la vitrina de productos

Mensaje recibido	Evento recibido	Actividad disparada	Mensaje enviado
	tocar	abrirPuerta, crear-DiálogoProductos	
	seleccionarM&S	validarAgregarM&S	validarAgregarM&S
agregarM&SVálido		cerrarPuerta	
	seleccionarSacarosa	validarAgregarSacarosa	validarAgregarSacarosa
agregarSacarosaVálido		cerrarPuerta	
	seleccionarAgarosa	validarAgregarAgarosa	validarAgregarAgarosa
agregarAgarosaVálido		cerrarPuerta	

Frigorífico

Cuadro 4.11: Descripción del Frigorífico

Mensaje recibido	Evento recibido	Actividad disparada	Mensaje enviado
	tocar	abrirPuerta, crear-DiálogoProductosFríos	
	seleccionarAuxina	validarAgregarAuxina	validarAgregarAuxina
agregarAuxinaVálido		cerrarPuerta	

Vaso de precipitados

Cuadro 4.12: Descripción del Vaso de precipitados

Mensaje recibido	Evento recibido	Actividad disparada	Mensaje enviado
entregarVaso		entregarVaso	
	adjuntar	adjuntarVaso	
	soltarVasoEnFregadero	validarSoltarVasoEnFregadero	validarSoltarVasoEnFregadero
soltarVasoEnFregaderoVálido		cambiarPosición	
soltarVasoEnFregaderoNoVálido		adjuntarVasoAvatar	
llenarVaso		llenarVaso, adjuntarVasoAvatar	
	soltarVasoEnAgitador	validarSoltarVasoEnAgitador	validarSoltarVasoEnAgitador
soltarVasoEnAgitadorVálido		cambiarPosición, bloquearAgitador	bloquearAgitador
soltarVasoEnAgitadorNoVálido		adjuntarVasoAvatar	
mezclar		girarAgua	
pararMezcla		pararGirarAgua	
moverMedio		vaciarVaso, eliminarVaso	

Fregadero

Cuadro 4.13: Descripción del Fregadero

Mensaje recibido	Evento recibido	Actividad disparada	Mensaje enviado
	tocar	validarLlenarVasoConAgua	validarLlenarVasoConAgua
llenarVasoConAguaVálido		fluirAgua	llenarVaso

Agitador

Cuadro 4.14: Descripción del Agitador

Mensaje recibido	Evento recibido	Actividad disparada	Mensaje enviado
bloquearAgitador		validarBloquearAgitador	validarBloquearAgitador
bloquearAgitadorVálido		bloquearAgitadorYPhmetro	bloquearPhmetro
bloquearAgitadorNoVálido		adjuntarVaso	adjuntarVaso
	tocar	validarEncenderAgitador	validarEncenderAgitador
encenderAgitadorVálido		mezclar	mezclar
	tocar	validarApagarAgitador	validarApagarAgitador
apagarAgitadorVálido		pararMezcla	pararMezcla
esAgitadorAvatar		verificarAvatar	esAgitadorAvatar, noEsAgitadorAvatar
moverMezclaABotes		desbloquearAgitador	validarDesbloquearAgitador

pH-metro

Cuadro 4.15: Descripción del pH-metro

Mensaje recibido	Evento recibido	Actividad disparada	Mensaje enviado
bloquearPhmetro		bloquearPhmetro	
	tocar	validarAjustarPh	validarAjustarPh

CAPÍTULO 4. SOLUCIÓN ADOPTADA

Botella para autoclavar

Cuadro 4.16: Descripción de la Botella para autoclavar

Mensaje recibido	Evento recibido	Actividad disparada	Mensaje enviado
entregarBotella		entregarBotella	
	adjuntar	adjuntarBotella	
	soltarBotellaEnMesaAgitador	validarSoltarBotellaEnMesaAgitador	validarSoltarBotellaEnMesaAgitador
soltarBotellaEnMesaAgitadorNoVálido		adjuntarBotellaAvatar	
soltarBotellaEnMesaAgitadorVálido		comprobarAgitadorAvatar	comprobarAgitadorAvatar
esAgitadorAvatar		cambiarPosición	
noEsAgitadorAvatar		adjuntarBotellaAvatar	
	soltarBotellaEnAutoclave	validarSoltarBotellaEnAutoclave	validarSoltarBotellaEnAutoclave
soltarBotellaEnAutoclaveNoVálido		adjuntarBotellaAvatar	
soltarBotellaEnAutoclaveVálido		cambiarPosición	bloquearAutoclave
	soltarBotellaEnCabina	validarSoltarBotellaEnCabina	validarSoltarBotellaEnCabina
soltarBotellaEnCabinaNoVálido		adjuntarBotellaAvatar	
soltarBotellaEnCabinaVálido		comprobarCabinaAvatar	comprobarCabinaAvatar
esCabinaAvatar		cambiarPosición, validarTiempoAtemperar	validarTiempoAtemperar
noEsCabinaAvatar		adjuntarBotellaAvatar	
	tocar	validarMoverMedioABotella	validarMoverMedioABotella
moverMedioABotellaVálido		llenarBotella, adjuntarBotellaAvatar	moverMedio
moverMezclaABotes		vaciarBotella, eliminarBotella, moverMedio	

Autoclave

Cuadro 4.17: Descripción del Autoclave

Mensaje recibido	Evento recibido	Actividad disparada	Mensaje enviado
bloquearAutoclave	validarBloquearAutoclave	validarBloquearAutoclave	
bloquearAutoclaveVálido		bloquearAutoclave	
	tocar	validarAutoclavar	validarAutoclavar
	tocar	validarEntregarBotellaAutoclavada	validarEntregarBotellaAutoclavada
entregarBotellaAutoclavadaVálido		entregarBotella	entregarBotella

Cabina de flujo laminar

Cuadro 4.18: Descripción de la Cabina de flujo laminar

Mensaje recibido	Evento recibido	Actividad disparada	Mensaje enviado
	tocar	validarEncenderCabina	validarEncenderCabina
encenderCabinaVálido		encenderCabina, validarTiempoFiltrarAire	bloquearEsterilizador, validarTiempoFiltrarAire
esCabinaAvatar		verificarAvatar	esCabinaAvatar, noEsCabinaAvatar
	tocar	validarApagarCabina	validarApagarCabina
apagarCabinaVálido		apagarCabina	

Esterilizador

Cuadro 4.19: Descripción del Esterilizador

Mensaje recibido	Evento recibido	Actividad disparada	Mensaje enviado
bloquearEsterilizador		bloquearEsterilizador	
	tocar	validarEsterilizarInstrumental	validarEsterilizarInstrumental

Bandeja de botes

Cuadro 4.20: Descripción de la Bandeja de botes

Mensaje recibido	Evento recibido	Actividad disparada	Mensaje enviado
	tocar	validarEntregarBandejaBotes	validarEntregarBandejaBotes
entregarBandejaBotesVálido		entregarBandejaBotesAvatar	
	adjuntar	adjuntarBandejaBotes	
	soltarBandejaBotesEnCabina	validarSoltarBandejaBotesEnCabina	validarSoltarBandejaBotesEnCabina
soltarBandejaBotesEnCabinaNoVálido		adjuntarBandejaBotesAvatar	
soltarBandejaBotesEnCabinaVálido		comprobarCabinaAvatar	comprobarCabinaAvatar
esCabinaAvatar		cambiarPosición	
noEsCabinaAvatar		adjuntarBandejaBotesAvatar	
	tocar	validarLlenarBotes	validarLlenarBotes
llenarBotesVálido		llenarBotes	moverMezclaABotes
	tocar	validarCortarPlanta	validarCortarPlanta
cortarPlantaVálido		colocarHojas	
	tocar	validarAdjuntarBandejaBotes	validarAdjuntarBandejaBotes
adjuntarBandejaBotesVálido		adjuntarBandejaAvatar	
	soltarBandejaBotesFitotrón	validarSoltarBandejaBotesFitotrón	validarSoltarBandejaBotesFitotrón
soltarBandejaBotesFitotrónNoVálido		adjuntarBandejaBotesAvatar	
soltarBandejaBotesFitotrónVálido		buscarBaseCercanaVacía	buscarBaseCercanaVacía
baseLibre		cambiarPosición	
baseBloqueada		adjuntarBandejaAvatar	
crecer		validarCrecerPlantas	validarCrecerPlantas
crecerPlantasVálido		crecerPlantas	

Bote con planta

Cuadro 4.21: Descripción del Bote con planta

Mensaje recibido	Evento recibido	Actividad disparada	Mensaje enviado
	tocar	validarEntregarBotePlanta	validarEntregarBotePlanta
entregarBotePlantaVálido		entregarBotePlanta	
	adjuntar	adjuntarBotePlanta	
	soltarBotePlantaEnCabina	validarSoltarBotePlantaEnCabina	validarSoltarBotePlantaEnCabina
soltarBotePlantaEnCabinaNoVálido		adjuntarBotePlantaAvatar	
soltarBotePlantaEnCabinaVálido		comprobarCabinaAvatar	comprobarCabinaAvatar
esCabinaAvatar		cambiarPosición	
noEsCabinaAvatar		adjuntarBotePlantaAvatar	
cortarPlantaVálido		cortarPlanta, eliminarBote	cortarPlanta

Fitotrón

Cuadro 4.22: Descripción del Fitotrón

Mensaje recibido	Evento recibido	Actividad disparada	Mensaje enviado
buscarBaseCercanaVacía		buscarBaseVacía	validarBloquearBaseFitotrón
bloquearBaseFitotrónNoVálido		baseBloqueada	baseBloqueada
bloquearBaseFitotrónVálido		baseLibre	baseLibre
tocarPuertaSalir		abrirPuerta	
tocarPuertaEntrar		abrirPuerta, crecerPlantas	crecer

Tutor

Cuadro 4.23: Descripción del Tutor

Mensaje recibido	Actividad disparada	Mensaje enviado
crearProtocoloPráctica1	crearNotaDelEstudiante	
validarAcción	validarAcción	acciónVálida, acciónNoVálida
validarAcciónTemporal	validarAcción, iniciarTemporizador	acciónVálida, acciónNoVálida

Este objeto, en especial su actividad de validar una acción, se explica en detalle en la siguiente sección.

4.6.2. Diseño del Componente Tutor

Para poder controlar la correcta realización de la práctica en el laboratorio se ha creado una primitiva central llamada “Tutor”, que es invisible para los avatares. El tutor tiene en su inventario un *script* que recibe por medio del canal 1 las peticiones de validación de las acciones que desea realizar el usuario, y después de validar dicha acción envía por el canal 2 el resultado de aprobación o de rechazo.

Para validar las acciones, el tutor contiene en su inventario una nota o *notecard* llamada *EstrategiaDeTutoria* donde se describen las secuencias correctas de acciones que cada estudiante debe realizar en el laboratorio, así como los mensajes de tutoría que mostrará el sistema al alumno cada vez que éste realice una acción. También tiene otra nota llamada *Bloqueos*, en la que se indica, para cada objeto en uso que no puede ser propiedad de un avatar, qué avatar lo está utilizando. De esta manera se evita que lo pueda utilizar otro avatar al mismo tiempo. Otra nota que utiliza el Tutor se llama *FaseActual*, para indicar en qué fase de la práctica se encuentra cada estudiante. Las acciones realizadas por cada estudiante se guardan en una nota con el nombre de su avatar, que representa el cuaderno de protocolo del estudiante.

A continuación, se van a describir con más detalle las notas que va a utilizar el Tutor.

Nota de la Estrategia de Tutoría

Representa las acciones que debe realizar el estudiante para culminar con éxito la práctica, así como las precondiciones de cada una de esas acciones. Asimismo, para cada acción se especifican una serie de mensajes que proporcionará el sistema al alumno a modo de guía. Cada línea describe la acción con varios elementos, que se detallan posteriormente, estos deben estar separados entre ellos por una barra vertical (|) de la siguiente forma:

```
FaseTarea|CodigoAcción|(1)/(0)-MensajeOK|MensajeAccionSiguiente|
Dependencias|(1)/(0)-MensajesErrorDependencias|
(1)/(0)-MensajesErrorOrdenDepend|Incompatibilidades|
(1)/(0)-MensajeErrorIncompatibilidad|Bloquea(1)Desbloquea(0)|
TiempoMaximo|TiempoMinimo|ValidarAccionYaRealizada|ValidarErroresFase|
CambiodeFase|
```

Ejemplo:

```
f1t11|encenderagitador|1-Agitador Encendido|Toca la puerta de la
vitrina o el frigorífico para agregar ingredientes a la mezcla,
no olvides ajustar el pH.|[f1t8]|1-No se ha llenado el vaso con
agua destilada| | | |1|0|0| | |1| |
```

Los elementos de la trama que describe a la acción son:

1. Código secuencial
2. Código descriptivo
3. Mensaje de aprobación
4. Mensaje de la siguiente acción a realizar
5. Códigos de las acciones de las que depende
6. Mensaje de error por falta de alguna de las dependencias
7. Mensaje de error por orden de las dependencias
8. Acciones incompatibles
9. Mensaje de error de incompatibilidades
10. Bandera de bloqueo o desbloqueo
11. Tiempo máximo
12. Tiempo mínimo
13. Bandera para validar repetición de dicha acción
14. Bandera para validar errores cometidos
15. Bandera que indica el final de una fase

El primer elemento es un código único, que indica la fase y el número de la acción dentro de ésta. En el ejemplo anterior se puede ver que la línea corresponde a la fase 1 tarea 11.

El “código descriptivo” es el que envía el objeto por el canal 1 para que el tutor identifique la acción y la valide, teniendo en cuenta el estado de la práctica en el que se encuentra el estudiante.

CAPÍTULO 4. SOLUCIÓN ADOPTADA

El “mensaje de aprobación”, precedido por un número (1 o 0) más un guión (-), se presenta al avatar cuando se ha validado correctamente la acción y dependiendo del número, es decir, si se antepone un 1, el mensaje debe ser mostrado, pero si es un 0 no se muestra.

El “Mensaje de la siguiente acción” se puede o no presentar como ayuda para el estudiante para darle una pista acerca de la siguiente acción que debe realizar.

Las “dependencias” determinan las acciones que deben haber sido realizadas antes de ejecutar la acción deseada, especificadas por medio de su código secuencial. Si sólo se depende de una acción anterior, simplemente se coloca su código entre un par de corchetes, pero si depende de más de una, se las debe agrupar separándolas entre sí por medio de guiones. Si no importa el orden en el que se hayan ejecutado las acciones, se las agrupa entre un par de corchetes, y si se necesita que hayan sido realizadas en un orden específico, se las agrupa entre un par de paréntesis.

Si no se ha ejecutado una de las dependencias se presenta el mensaje de error correspondiente definido en el sexto elemento. A este mensaje se le antepone un número (1 o 0) más un guión (-), para indicar si el error permite (0) o no (1) seguir con el proceso de la práctica. Si no se permite seguir con la práctica, se muestra dicho mensaje al avatar, y si se permite seguir, el mensaje no se muestra y se registra en la nota del cuaderno de protocolo del avatar que corresponda. Si existe más de una dependencia, los mensajes con su número antepuesto, se separan entre ellos con una barra inclinada (/) en el mismo orden en el que se encuentran sus dependencias asociadas.

El “mensaje de error por orden de dependencia” debe presentarse al avatar si se han ejecutado todas las acciones de las que depende la acción actual, pero en un orden incorrecto. De igual manera que en el tipo de mensaje anterior, se antepone un número y un guión para especificar si se permite seguir o no al estudiante con la práctica por el error cometido, e igualmente si se presenta o no dicho mensaje al avatar.

La figura 4.34 indica la secuencia de acciones de una parte del proceso de la práctica y explica las dependencias de la última acción. Todas estas acciones tienen sus dependencias, pero la figura solo ilustra las dependencias de la última acción a modo de muestra. Esta secuencia representa la creación de la mezcla para el medio de micropropagación de las plantas, que comienza al encender el agitador y termina al apagar el mismo. En las dependencias de esta última acción se puede ver que el orden de las acciones es importante a excepción de dos acciones: agregar Murashige&Skoog y agregar sacarosa a la mezcla. Éstas no importa en qué orden se realicen, pero las dos tienen que haber sido realizadas después de encender el agitador y antes de ajustar el pH.

En las “acciones incompatibles” se especifican aquellas acciones, separadas por un guión, que no pueden haber sido ejecutadas antes de la actual. Es decir, si ya existe una de éstas en el cuaderno de protocolo del avatar, se presentará el mensaje

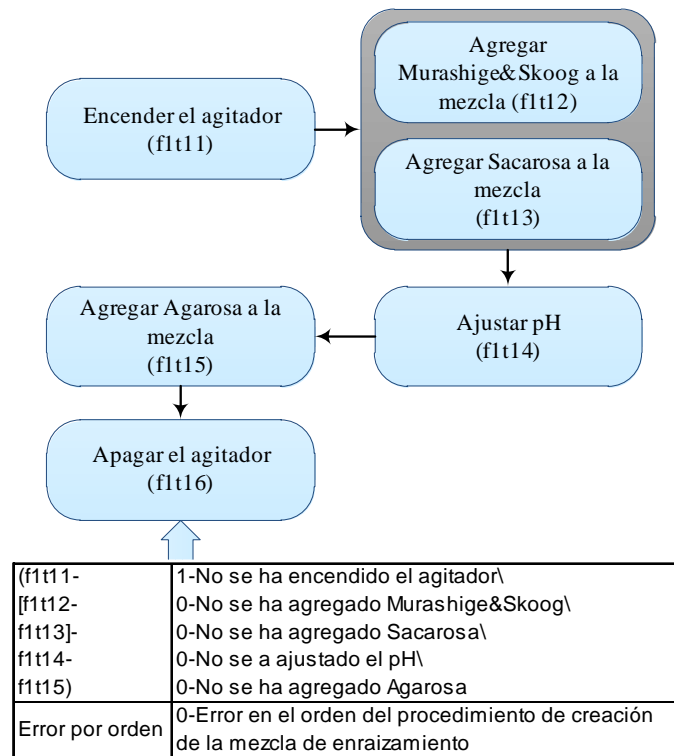


Figura 4.34: Especificación de dependencias del tutor

de error definido en el noveno elemento. De igual manera, si son más de una las actividades incompatibles se las debe separar por un guión, y sus mensajes de error van precedidos por un número que determina el tratamiento del error, y separados por una barra inclinada (/). En el siguiente ejemplo, se puede ver que la acción flt28 (*soltarbandeja*) no se puede ejecutar si ya se ha realizado la acción flt33 (*apagarcabina*).

```
flt28|soltarbandeja|0-Bandeja en cabina|Toca la bandeja para verter el
contenido de la botella en los botes.|[flt22-flt23]|0-No se ha encendido
la cabina de flujo\0-No se ha esterilizado por completo el aire| |flt33|
0-Se ha apagado la cabina antes de terminar el proceso.| |0|0| | |1| |
```

La “bandera de bloqueo” indica si esta acción bloquea al objeto que la envió (1) o si lo desbloquea (0), y por tanto es necesario actualizar la nota de bloqueos. Si la acción no bloquea ni desbloquea ningún objeto, se deja el campo en blanco.

El “tiempo máximo” es un valor entero, que indica el tiempo máximo en segundos que puede transcurrir hasta que se ejecute la siguiente acción. Mientras que el “tiempo mínimo”, que de la misma forma es un valor entero, determina el tiempo mínimo en segundos que se debe esperar para poder realizar la siguiente acción. El siguiente ejemplo indica que para que la botella esté atemperada se necesita esperar 20 segundos, es decir, la acción *enfriarbotella* se anotará en la nota del cuaderno de

CAPÍTULO 4. SOLUCIÓN ADOPTADA

protocolo después de 20 segundos de haber sido enviado el mensaje al tutor. Este ejemplo también muestra que la siguiente acción, que es *agregaríaa*, se tiene que ejecutar dentro de los 240 segundos posteriores a haberse enfriado la botella.

```
f1t25|enfriarbotella|1-Botella atemperada|Agrega ácido indolacético  
del frigorífico.|[f1t24]| | | | |240|20|0-Ha pasado demasiado tiempo,  
el medio se ha solidificado dentro de la botella, debes comenzar de  
nuevo.| |1| |
```

La “bandera de validar acción ya realizada” indica si dicha acción puede ser repetida varias veces (0) o no (1). Por ejemplo la siguiente acción no la puede realizar más de una vez.

```
f0t5|lavarsemanos|1-Manos limpias|Toca la caja de guantes de latex  
de la poyata para obtener unos.|[f0t1]|0-No se ha tomado una bata|  
| | | |0|0| | |1| |
```

La “bandera de validar errores de fase” determina con un número (1) si la acción actual debe verificar en el cuaderno de protocolo del avatar los errores que no han sido comunicados al estudiante durante la práctica y se han registrado en dicha nota.

El “cambio de fase” indica con un número (1) si la acción es la última de una fase para que el tutor pueda cambiar la fase en la que se encuentra el avatar.

Nota de Bloqueos

Contiene los nombres de los objetos que pueden ser bloqueados seguidos del UUID del avatar que los está utilizando. Cada línea en la *notecard* tiene el nombre del objeto y el UUID del avatar que lo está utilizando separados por una barra vertical (|), si no lo está utilizando ningún avatar solo contiene el nombre del objeto seguido de la barra. A continuación se muestra un ejemplo de una nota de bloqueo con dos líneas, en la primera, el objeto se encuentra ocupado y en la segunda no:

```
Agitador\_1|058a9fa1-b451-473e-8f40-3c4a55f5cf58  
Agitador\_2|
```

Nota del Cuaderno de protocolo

Esta nota se crea al inicio de la práctica con el nombre del avatar y contiene todas las acciones que ha realizado el estudiante. Cada línea representa una acción validada por el tutor y realizada por el avatar, sus componentes se separan por una barra vertical (|) y son los siguientes:

- Código secuencial de la acción
- Código de la acción

- Tiempo Unix en el que se realizó la acción (solo para acciones con tiempo)
- Código secuencial de las dependencias que permiten seguir con la práctica que no se han ejecutado.
- Código secuencial de las acciones incompatibles que se han realizado, pero que permiten seguir con la práctica.

Nota de la Fase Actual

Contiene el UUID de los avatares seguidos de una barra vertical (|) y la fase en la que se encuentra. Cada vez que se crea un cuaderno de protocolo se crea también una línea en esta *notecard* con el UUID del avatar y con la fase uno (f1). Posteriormente si se ejecuta una acción que provoca un cambio de fase, aumenta el número de la fase. Un ejemplo de configuración es el siguiente:

```
058a9fa1-b451-473e-8f40-3c4a55f5cf58|f1
```

Funcionamiento del Tutor

El tutor escucha siempre por el canal 1 los mensajes de los objetos que indican la acción a validar, deben tener el siguiente formato:

Operación|Key del Avatar|Acción|Nombre del Objeto

Las operaciones pueden ser *crearLibro*, *borrarFase*, *validar* o *borrar*. En el caso de la operación *crearLibro* no es necesario enviar nada en el campo acción, por lo tanto se puede mandar un espacio en blanco.

Cuando el tutor recibe el mensaje, lo procesa de la siguiente manera. Si la operación es *crearLibro*, genera una *notecard*, cuyo nombre es el mismo del avatar que está realizando la acción. Si es *borrarFase*, elimina la última fase realizada de la nota del usuario para que pueda volver a hacerla. Si la fase a eliminar es la primera, se borra la *notecard* del cuaderno de protocolo. La operación *validar*, verifica si la acción es válida y en caso afirmativo, modifica el cuaderno de protocolo del avatar para agregar la nueva línea, y si la operación es *borrar*, la elimina de esta *notecard*, debido a un error reportado por el objeto.

Para validar la acción, el tutor realizar varios pasos detallados a continuación:

1. Se obtiene la línea de la estrategia de tutoría de dicha acción para la fase en la que se encuentra el estudiante.
2. Se verifica si el objeto sobre el que se va a realizar la acción no está bloqueado por otro avatar.
3. Si está bloqueado.

- a) Se muestra un mensaje de error al usuario y se pasa al paso 19.
4. Se revisa si dicha acción ha sido realizada anteriormente.
5. Si se ha realizado la acción.
 - a) Se muestra un mensaje de error y se pasa al paso 19.
6. Se obtienen de la nota de estrategia de tutoría sus dependencias.
7. Por cada una de sus dependencias se valida que éstas hayan sido realizadas anteriormente.
8. En caso de no haber sido realizada una dependencia se verifica si ésta bloquea o no el proceso.
 - a) Si bloquea el proceso, se muestra el correspondiente mensaje de error y se pasa al paso 19.
 - b) Si no bloquea el proceso, se guarda en la lista de errores de dependencias.
9. Se verifica si existen acciones realizadas que sean incompatibles con la actual.
10. En caso de haber sido realizada una de las acciones incompatibles se verifica si ésta bloquea o no el proceso.
 - a) Si bloquea el proceso, se muestra el correspondiente mensaje de error y se pasa al paso 19.
 - b) Si no bloquea el proceso, se guarda en la lista de errores de incompatibilidades.
11. Si la acción está configurada con la bandera de bloqueo.
 - a) Se anota el id del avatar junto al nombre del objeto con el que se está interactuando en la *notecard* de bloqueos.
12. Si la acción está configurada con la bandera de verificar errores no bloqueantes, se obtienen los errores cometidos anteriormente desde la *notecard* del cuaderno del alumno.
13. Si la acción tiene la bandera de cambio de fase, se aumenta el número de la fase en la que se encuentra el estudiante en la nota correspondiente
14. Si se tiene configurado para dicha acción el tiempo mínimo o el tiempo máximo, se verifica si no está ya ejecutado el temporizador para otro avatar.
 - a) Si se ha ejecutado, se guarda la diferencia entre el tiempo anterior y el nuevo, para inicializarlo con éste al momento de terminar dicho temporizador.
 - b) Si no se ha ejecutado, se inicializa el temporizador con el valor de tiempo configurado.

- c) Si está configurado el tiempo mínimo, se cambia el código de la acción añadiendo al código de la acción un sufijo para poder identificarla como una acción de este tipo hasta que se cumpla el temporizador, y se anota el tiempo actual, junto al código, precedido por una barra (|).
 - d) Si está configurado el tiempo máximo, se anota el tiempo actual, junto al código, precedido por una barra (|).
15. Se presenta el mensaje de la siguiente acción si existiera.
 16. Se guarda la acción en la *notecard* del cuaderno de protocolo.
 17. Si no han existido errores se envía por el canal 2 un mensaje con el código de la acción anteponiendo un 1.
 18. Si ha existido algún error que no bloquee el proceso se envía por el canal 2 un mensaje con el código de la acción anteponiendo un -1.
 19. Si ha existido algún error que bloquee el proceso se envía por el canal 2 un mensaje con el código de la acción anteponiendo un 0.

El temporizador tiene un tratamiento especial, mencionado a breves rasgos anteriormente. Cuando se ejecuta este temporizador se procesan los siguientes pasos:

1. Por cada *notecard* de cuaderno de protocolo (o sea por la nota de cada estudiante), se obtiene la primera acción con sufijo temporal.
2. Se verifica si se ha cumplido el tiempo.
 - a) Si se ha cumplido el tiempo y tiene el sufijo de tiempo mínimo, se guarda la acción como definitiva y se envía el mensaje de ayuda al estudiante con el siguiente paso.
 - b) Si se ha cumplido el tiempo y no tiene el sufijo de tiempo mínimo quiere decir que es una acción con tiempo máximo, por lo cual se elimina la acción de la *notecard* si no está realizada ya la acción que dependa de ésta.
 - c) Si existe otra acción temporal posterior, en cualquier *notecard* de cualquier otro estudiante, se vuelve a iniciar el temporizador con el tiempo restante.

4.7. Diseño de la interacción entre objetos

Los siguientes diagramas de secuencias muestran las interacciones entre el avatar y los objetos que se encuentran en el entorno virtual. El avatar se encuentra siempre al inicio de todos los diagramas y las flechas que salen de él representan los eventos o acciones que puede ejecutar sobre los objetos. Una flecha que sale de un objeto A

y llega a un objeto B denota una actividad que es ejecutada por el objeto A y que en algún momento de su ejecución manda un mensaje al objeto B. En cambio, una flecha que sale de un objeto A y llega al mismo objeto denota una actividad que es ejecutada por el objeto A y no manda ningún mensaje a ningún objeto.

A continuación, se modelarán las principales tareas del protocolo detallado en la sección 4.4.

El proceso se inicia cuando el estudiante selecciona la práctica, se pone bata y lee los fundamentos y protocolo de dicha práctica. Este proceso se encuentra representado en las figuras 4.35, 4.36 y 4.37.

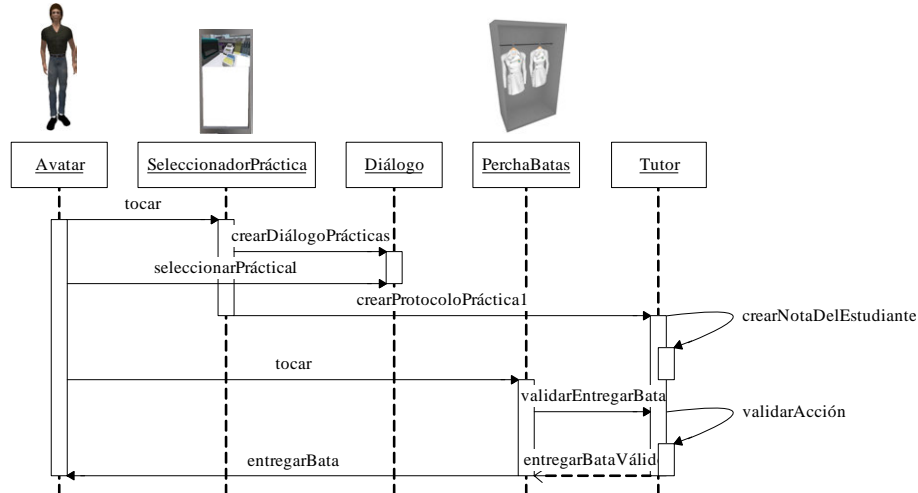


Figura 4.35: Diagrama de secuencia de la creación del inicio de la práctica (i)

Las figuras 4.38, 4.39 y 4.40 representan la creación del medio de enraizamiento para las plantas. Esta tarea comienza al tomar un vaso de la vitrina de instrumental y termina al parar la mezcla en el agitador.

Después de crear el medio de enraizamiento es necesario autoclavarlo y pasar el medio a los botes en los que crecerán las plantas, esto muestran las figuras 4.41, 4.42, 4.43 y 4.44.

Después de tener el medio en los botes de vidrio se puede micropropagar el material vegetal desde una planta inicial y colocar pequeños brotes con un pedazo de tallo y una hoja en cada bote, para finalmente hacer crecer las plantas en el fitotrón. Este proceso se detalla en las figuras 4.45 y 4.46.

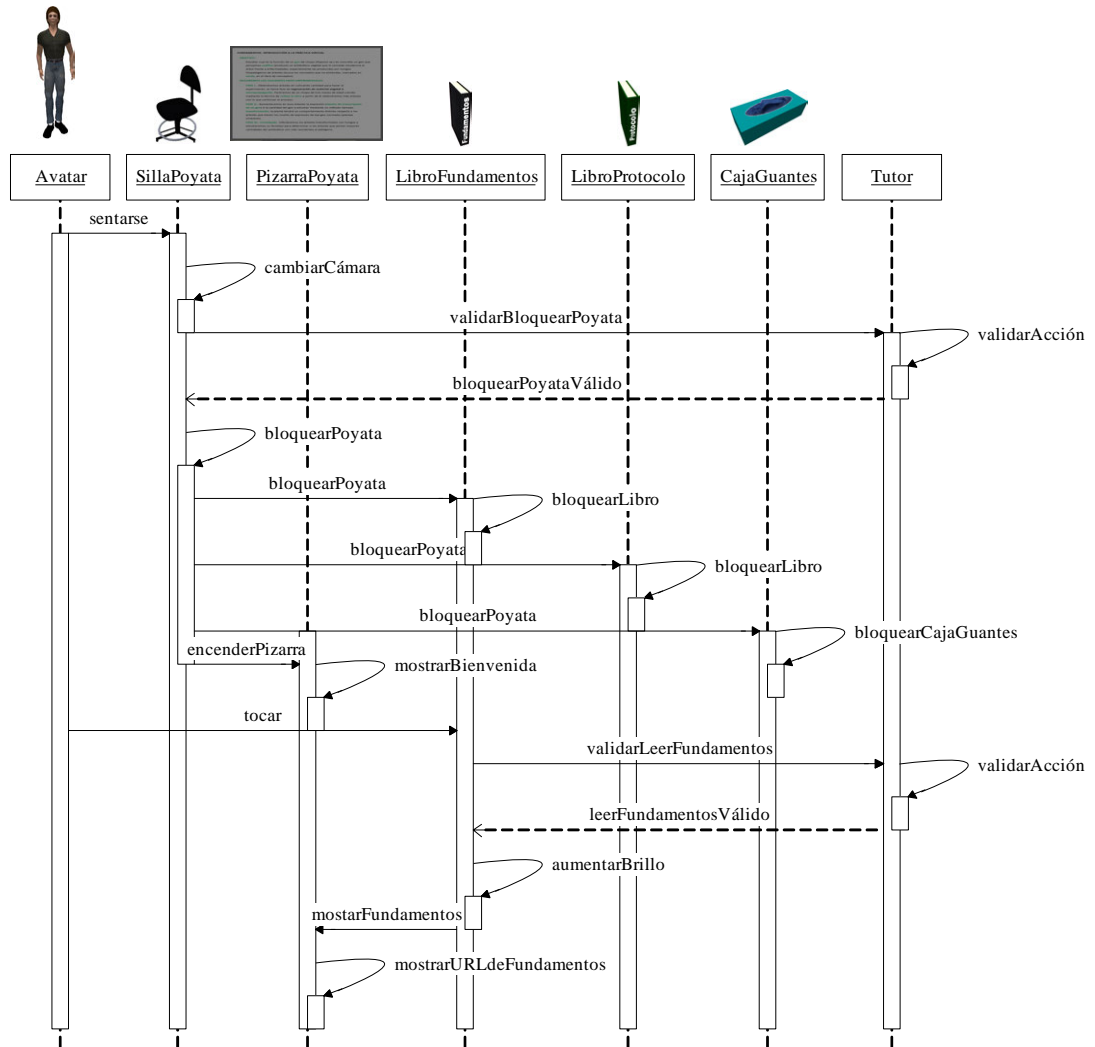


Figura 4.36: Diagrama de secuencia del inicio de la práctica (ii)

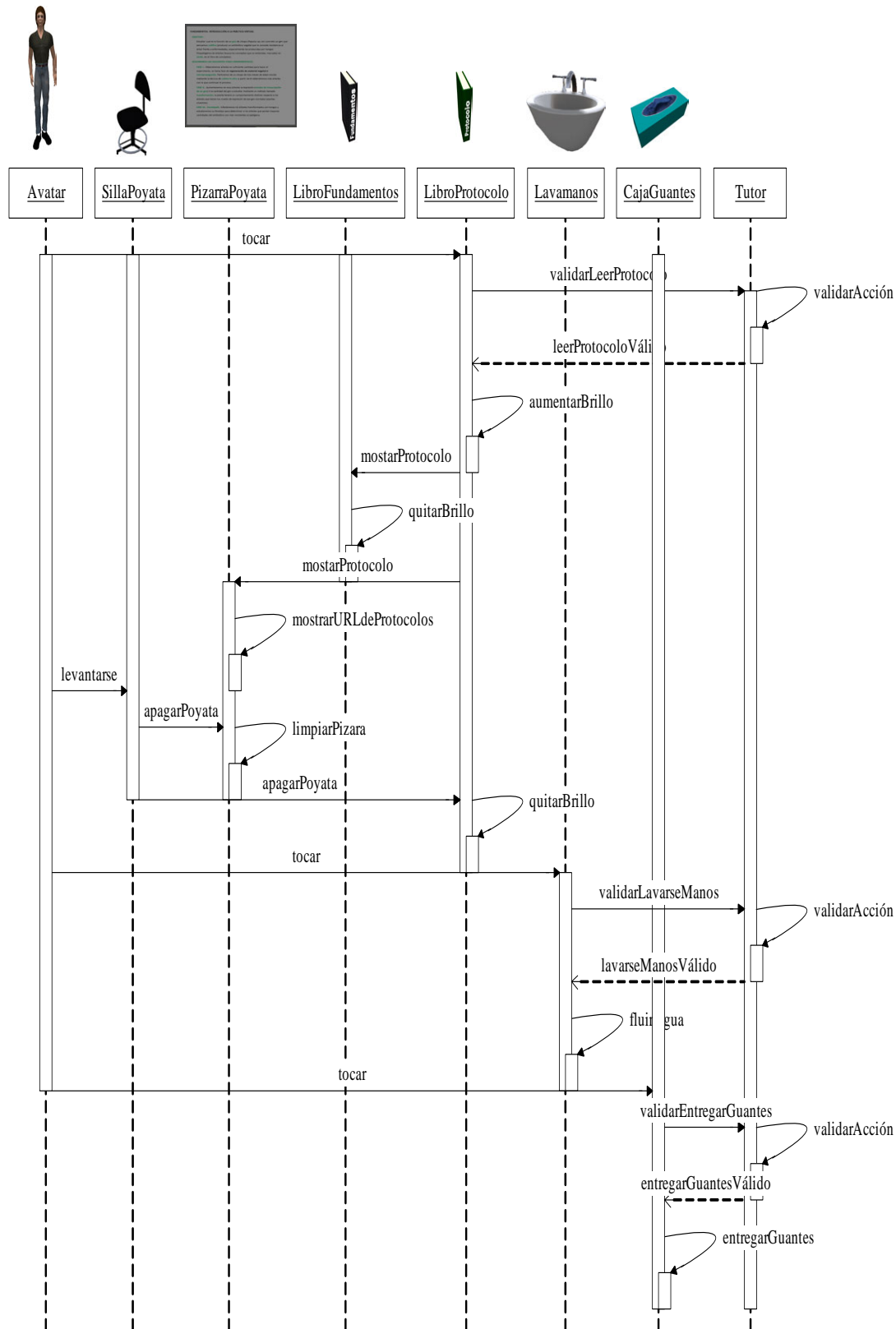


Figura 4.37: Diagrama de secuencia del inicio de la práctica (iii)

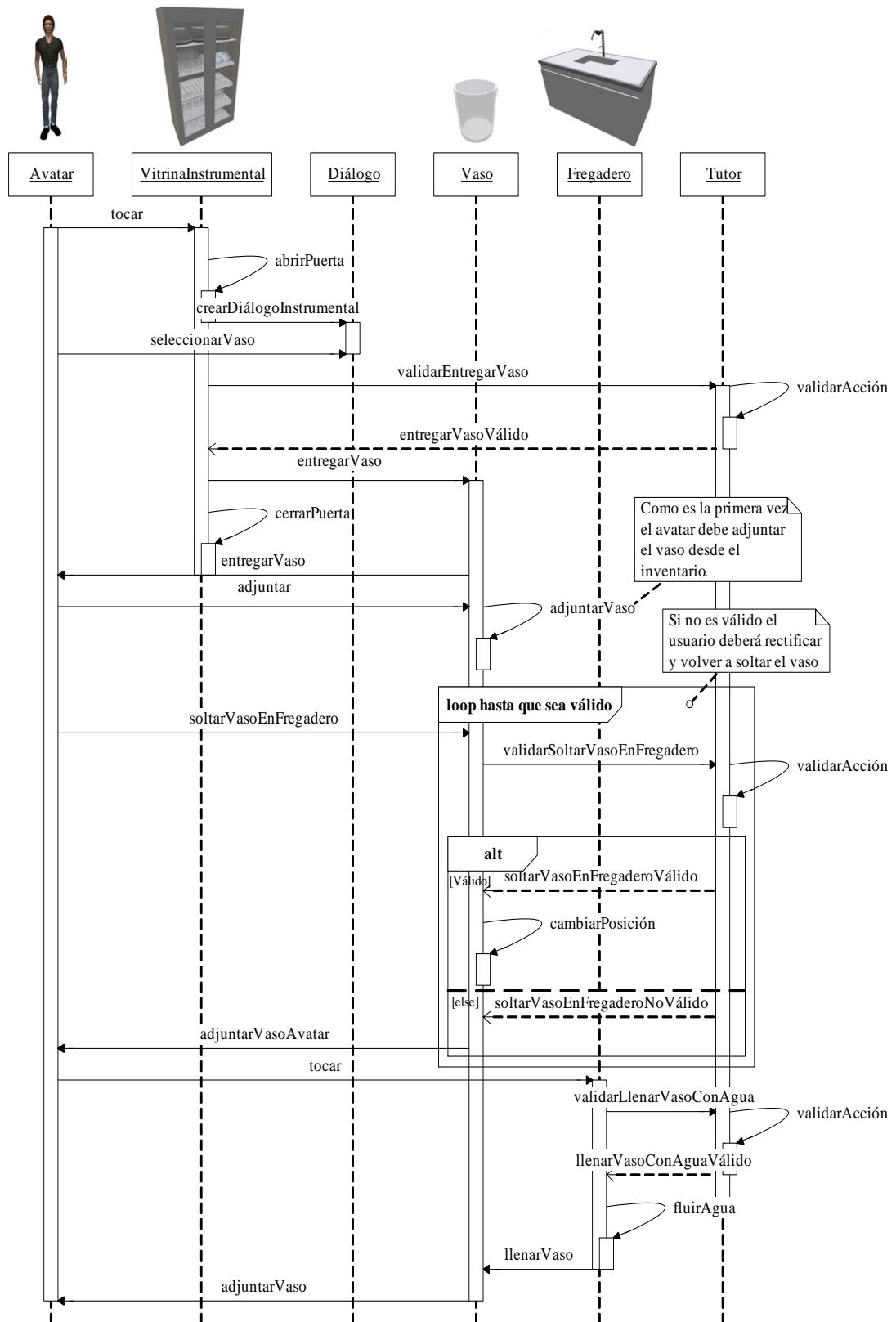


Figura 4.38: Diagrama de secuencia de la creación del medio (i)

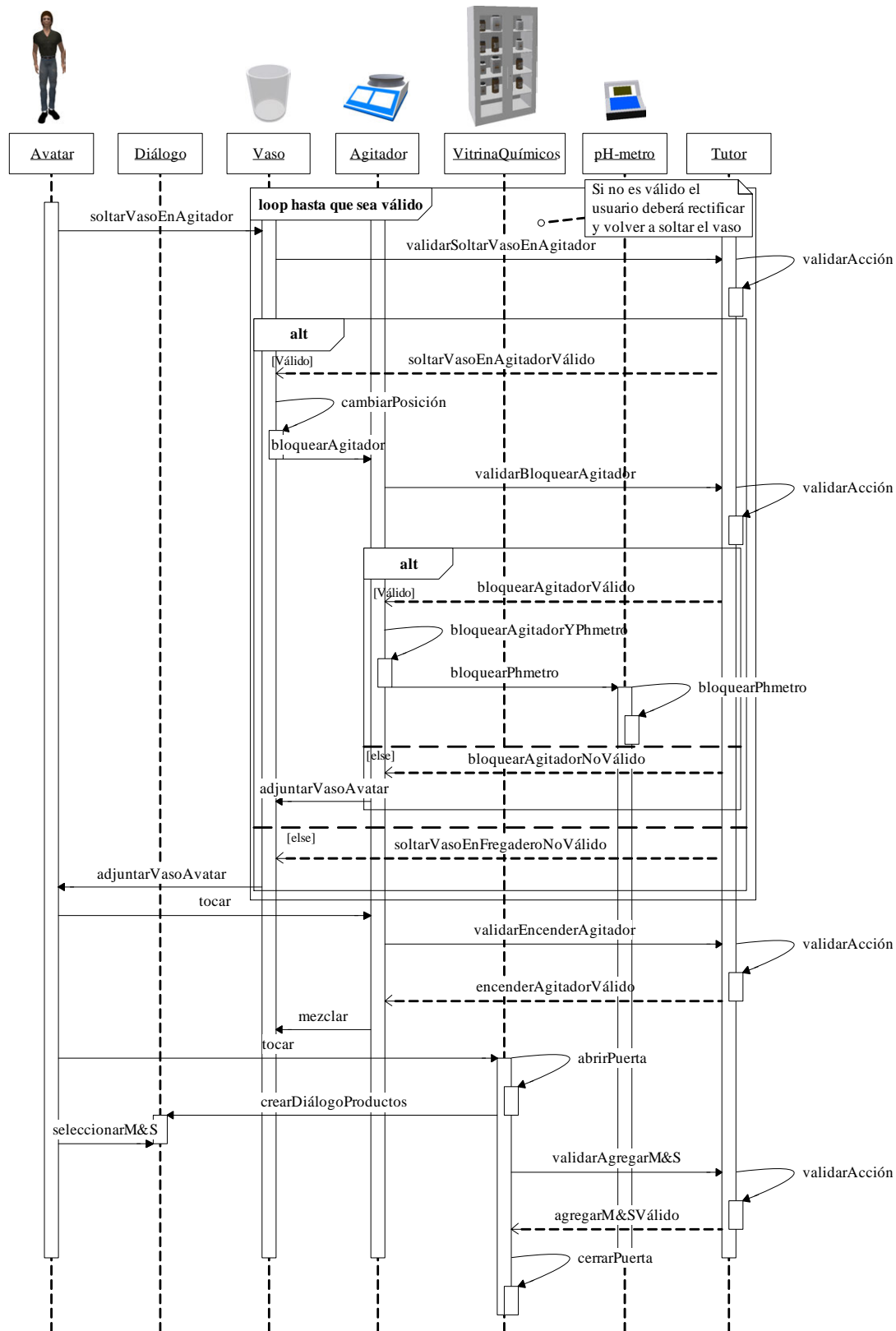


Figura 4.39: Diagrama de secuencia de la creación del medio (ii)

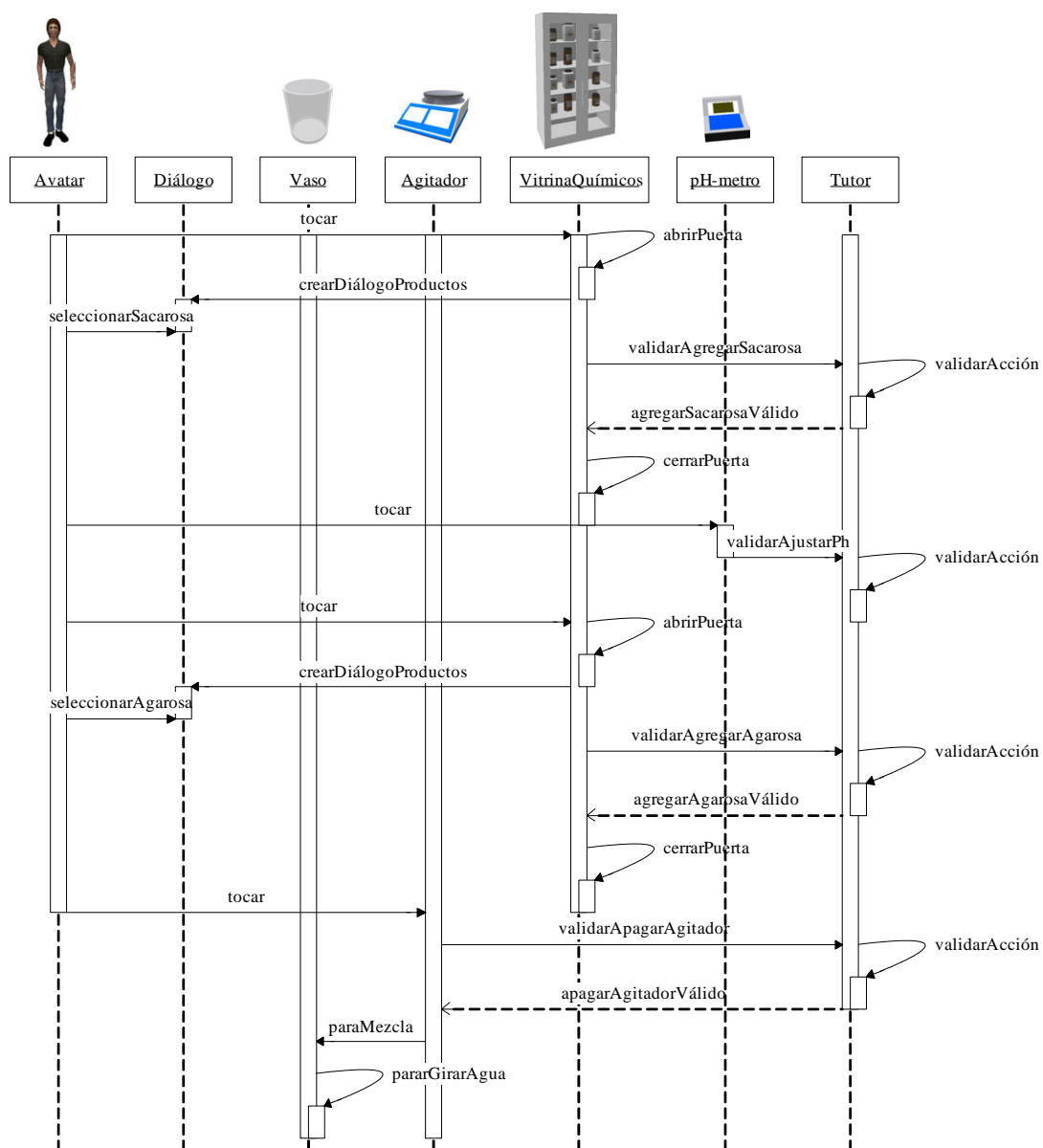


Figura 4.40: Diagrama de secuencia de la creación del medio (iii)

CAPÍTULO 4. SOLUCIÓN ADOPTADA

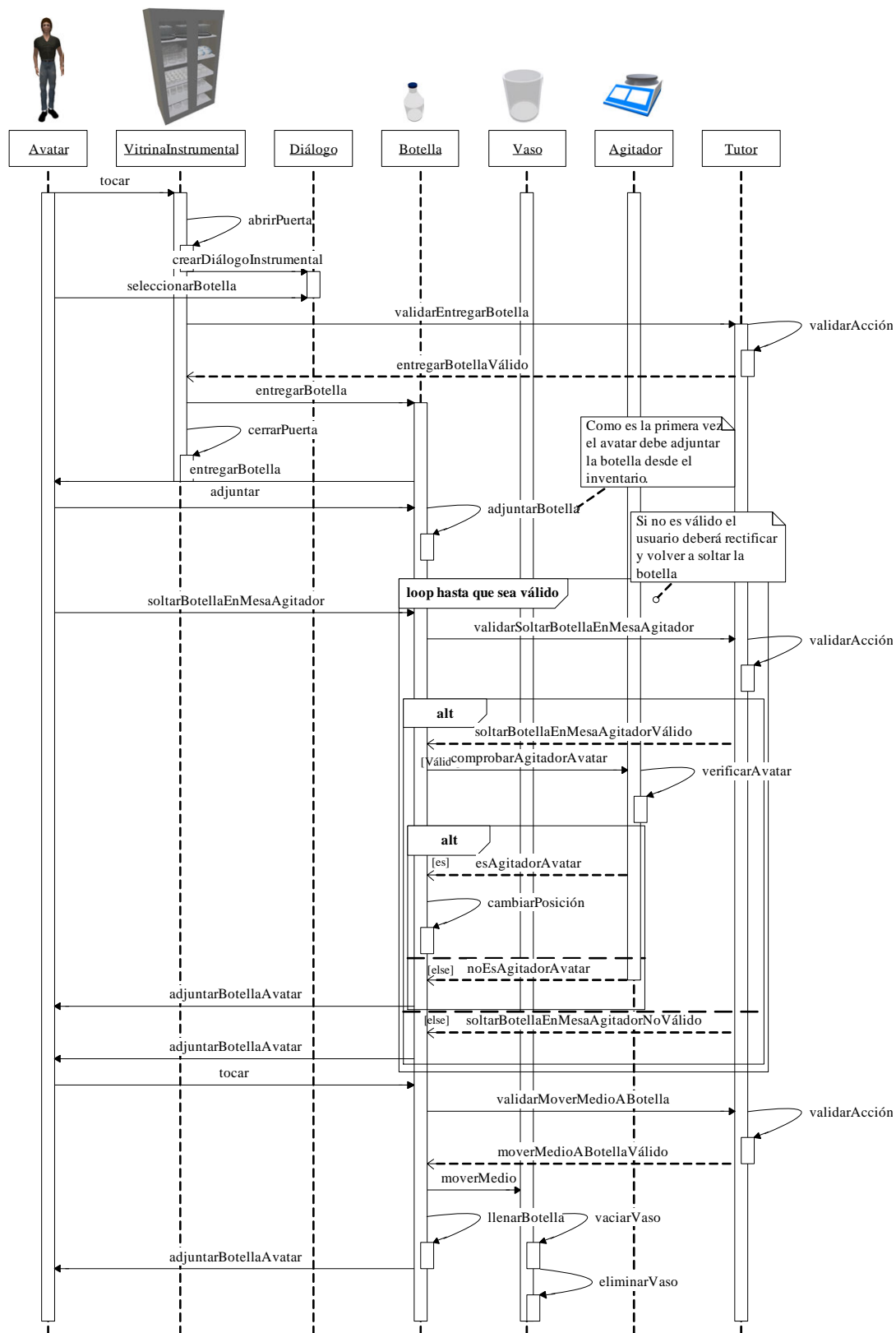


Figura 4.41: Diagrama de secuencia del proceso de mover el medio a botes de vidrio
(i)

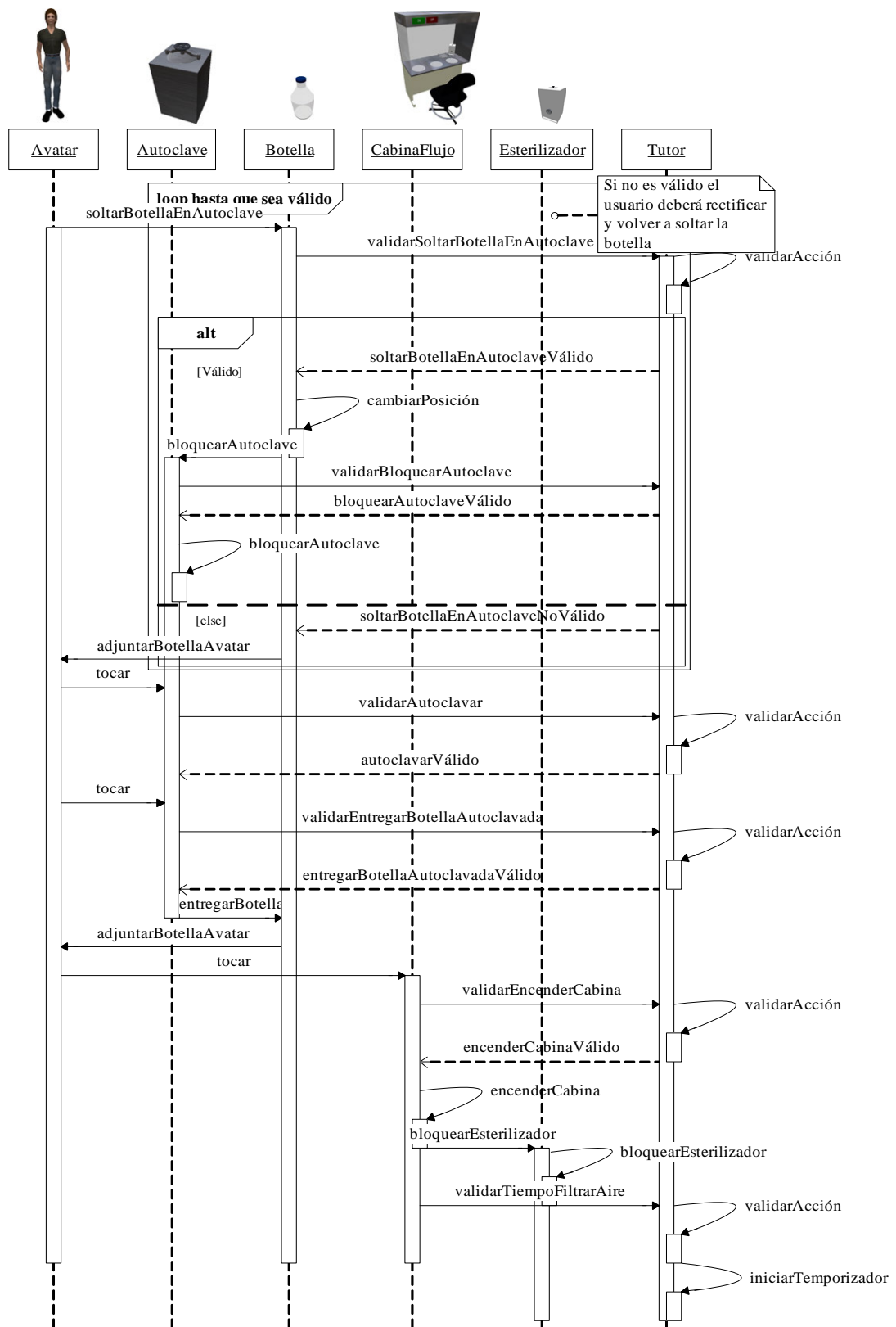


Figura 4.42: Diagrama de secuencia del proceso de mover el medio a botes de vidrio (ii)

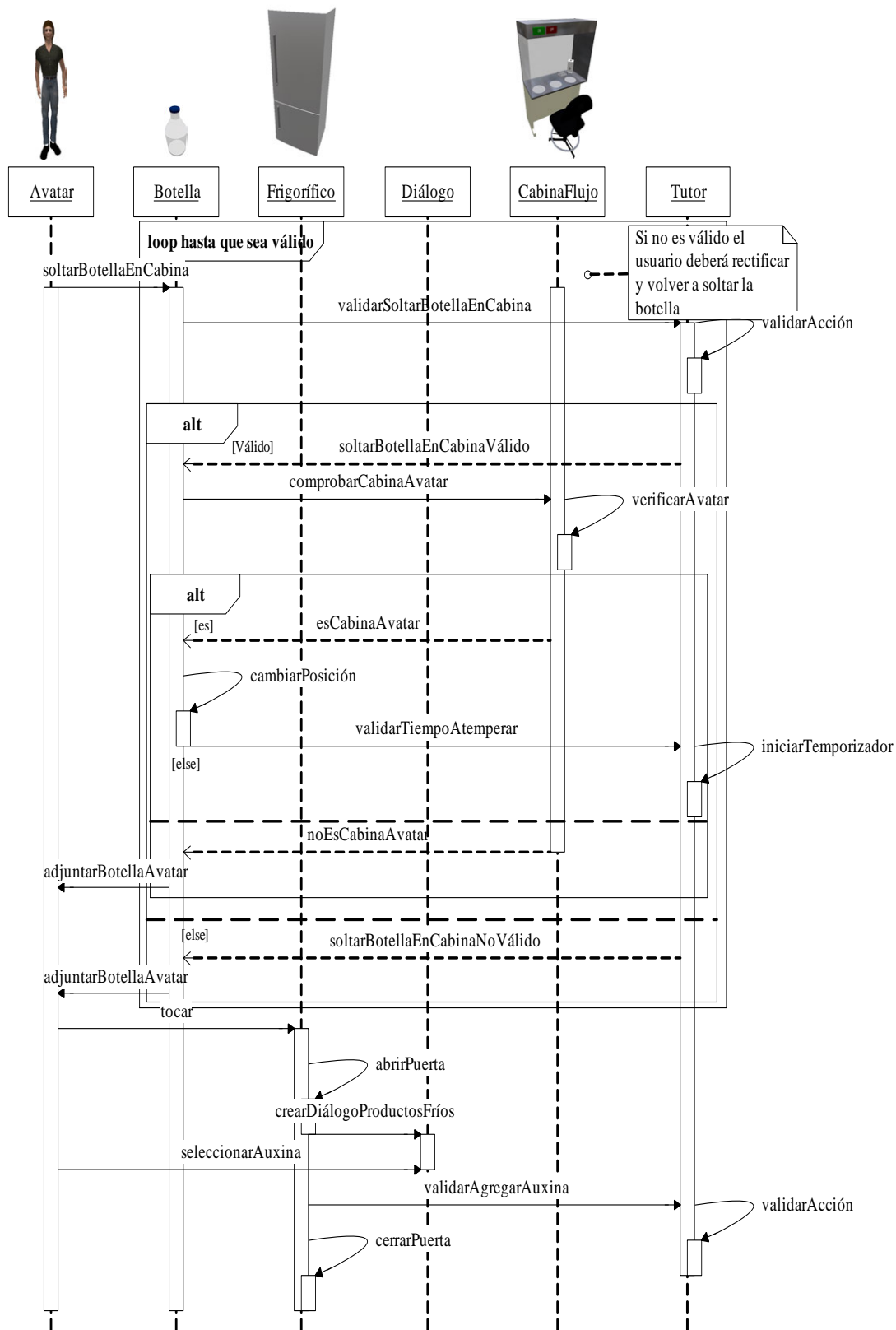
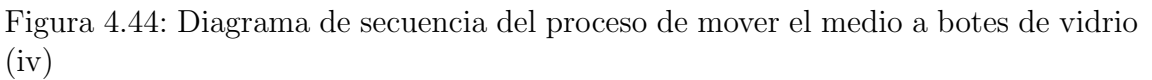


Figura 4.43: Diagrama de secuencia del proceso de mover el medio a botes de vidrio (iii)



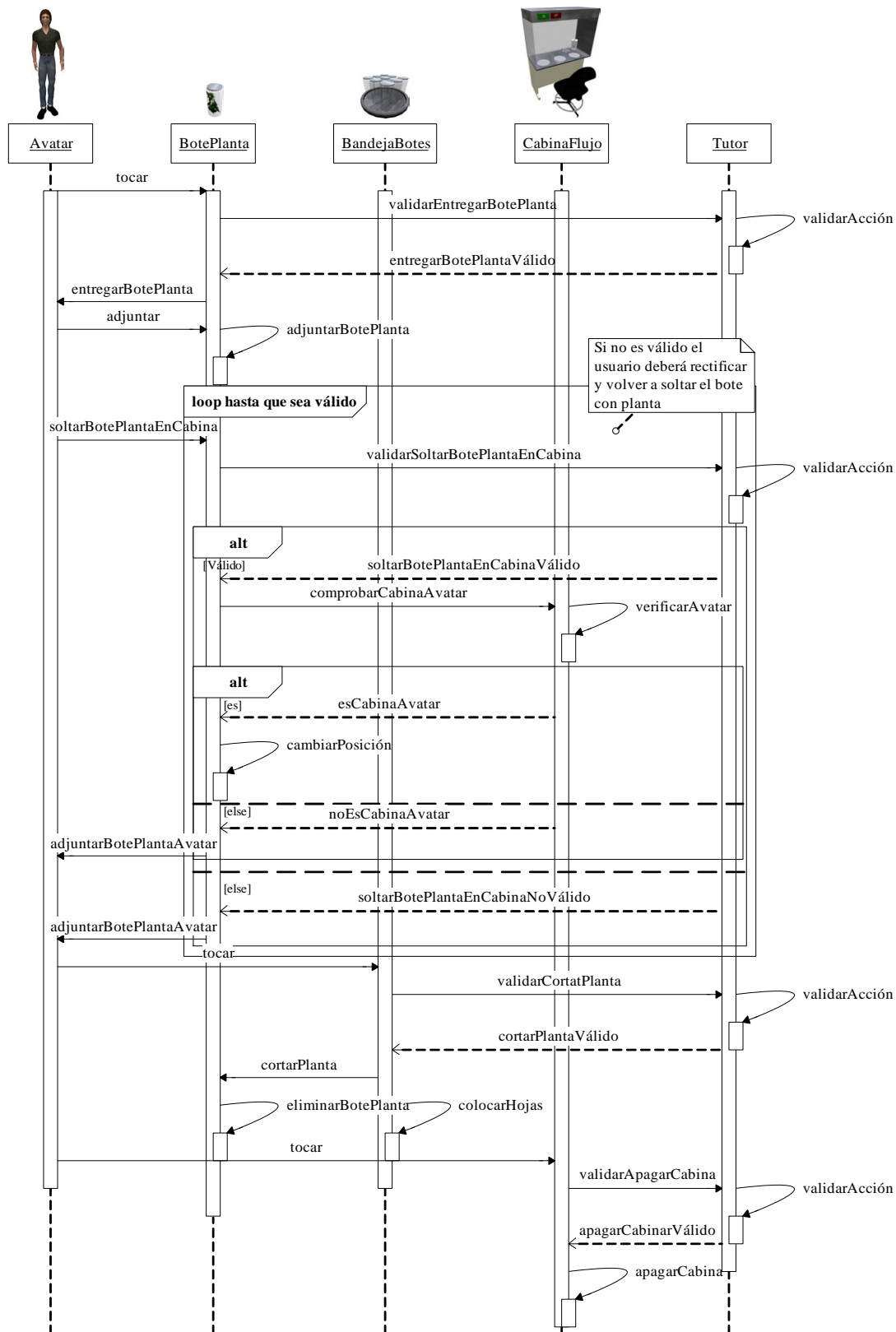


Figura 4.45: Diagrama de secuencia del crecimiento de material vegetal (i)

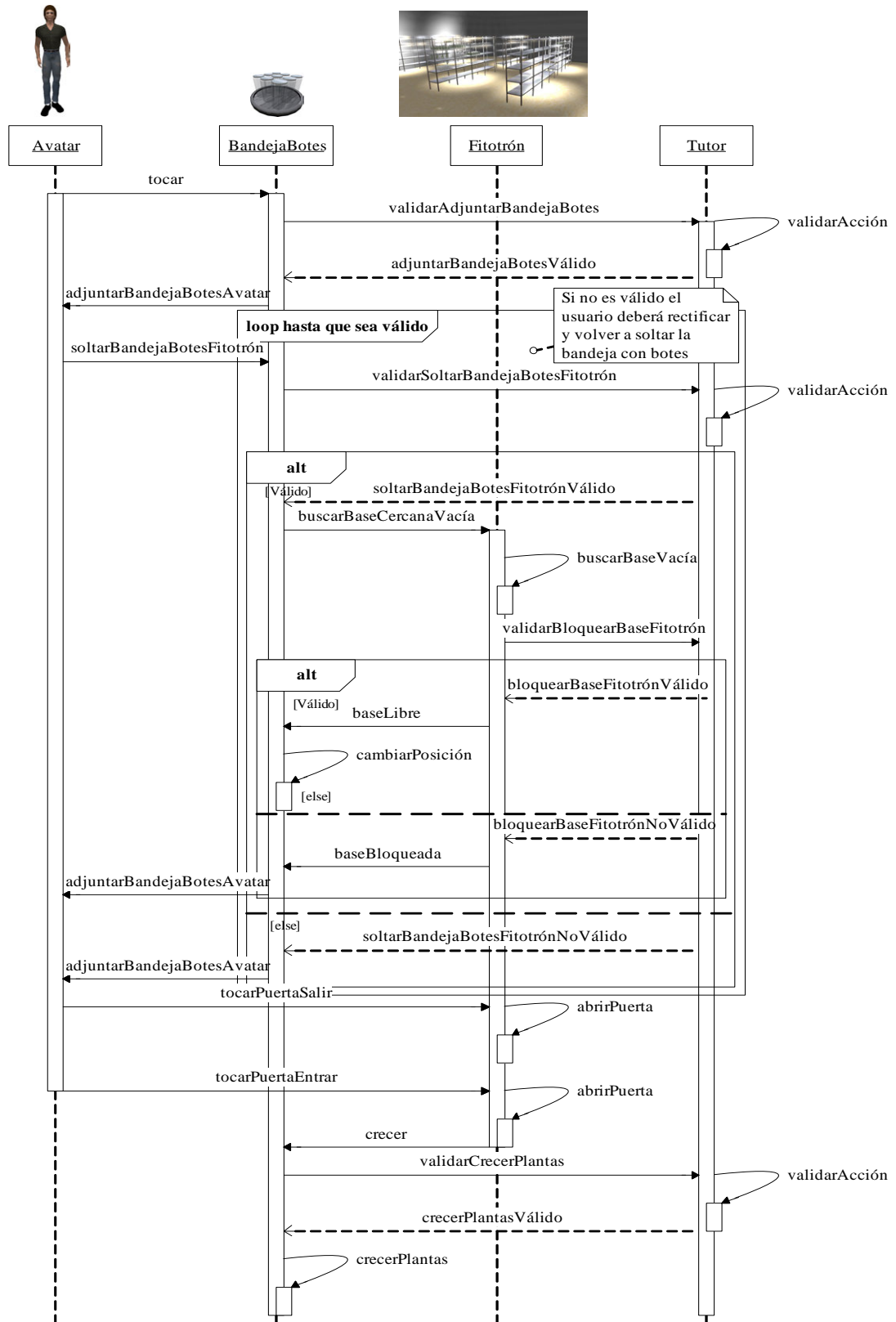


Figura 4.46: Diagrama de secuencia del crecimiento de material vegetal (ii)

Capítulo 5

Conclusiones y trabajos futuros

Este trabajo presenta los beneficios de los entornos virtuales para potenciar la educación mediante el uso de tecnologías *online*. En un mundo donde el sistema educativo recibe cada vez más estudiantes, y en el que los Estados disponen de menos recursos para educación, los espacios virtuales educativos presentan ventajas frente al modelo tradicional, dado que disminuyen costes, y además su cobertura puede ser mayor que la de un aula de clase, ya que pueden llegar a más personas y en distintas ubicaciones. Otra de sus ventajas radica en que este tipo de plataformas son espacios colaborativos de aprendizaje continuo y participativo, lo cual puede cambiar el modelo de academia clásica, en el que el estudiante participaba poco y solo se limitaba a hacer lo que le enviase el profesor para o bien aprender u obtener una calificación. En cambio, en este tipo de entornos, el alumno intercambia información, experiencias y conocimientos con otras personas, lo que fomenta un mayor interés, además de un mayor desarrollo de competencias.

Un elemento importante, aunque poco mencionado en la literatura del campo informático, es el factor lúdico que conllevan este tipo de entornos. No es lo mismo aprender en una simulación virtual con varias personas participando simultáneamente que en una clase formal, las cuales, en algunos casos, pueden resultar ser de poco interés para el alumno, y su asistencia se puede convertir en un mero formalismo para aprobar una asignatura, por lo que al final el aprendizaje es de menor valor.

Cabe destacar que a través de las interacciones educativas mediadas por las tecnologías, los estudiantes tienen mayor confianza en participar, equivocarse y rectificar. En este caso, gracias al desarrollo de este laboratorio de biotecnología, el alumno tiene la seguridad de no dañar el instrumental o material costoso. Además, si se equivoca, puede rectificar, lo que no ocurre en un laboratorio real, ya que el fallo en un experimento representa en la mayoría de los casos una significativa suma de dinero perdido o la posibilidad de sufrir algún daño físico.

Centrados en la implementación del laboratorio, es importante resaltar a Open-Simulator como una buena herramienta para el desarrollo de entornos virtuales, ya

que es gratuito y de código abierto. Además, al estar basada en Second Life, nos podemos aprovechar de su madurez en lo que se refiere fundamentalmente al desarrollo de los visores y del abundante material disponible, tales como ejemplos de *scripts* y objetos tridimensionales ya contruidos. Asimismo, se cuenta con la ayuda de una amplia comunidad de desarrolladores. Gracias a todo esto se pueden desarrollar entornos virtuales con un alto nivel de realismo. Por otra parte, una de sus desventajas radica en que hasta la fecha no existe una versión estable del servidor, y esto implica problemas de fiabilidad.

Al ser una práctica que debe seguir un procedimiento riguroso, se vio la necesidad de crear un objeto, al que se llamó “tutor”, que valide los pasos que va realizando el estudiante, y que se aprovechó también para ayudar al alumno indicándole cuál es su siguiente acción a realizar. Su programación es genérica, es decir, puede adaptarse a otros procesos de laboratorio con una fácil configuración, sin necesidad de conocimientos de programación, siendo así una herramienta que se podría utilizar en laboratorios virtuales de otras ramas de la ciencia.

Durante el desarrollo del entorno virtual, se han ido encontrando algunos problemas que no suelen presentarse en los sistemas de software comunes. Uno de estos problemas tiene que ver con la interacción con el sistema, ya que las acciones que se pueden realizar con los objetos 3D son más que las que se puede realizar en un sistema típico de ingreso de datos con una interfaz gráfica 2D. Esto se solucionó con la validación de las acciones por parte del tutor.

Otro problema que se descubrió fue con respecto a la propiedad de multiusuario del sistema, pues los avatares podían interferir lo que otro estaba realizando, lo cual al bloquear los objetos por usuario no volvió a suceder. Además, los mensajes que enviaba por un canal un objeto propiedad de un avatar, como el vaso de precipitados, podían llegar a otro avatar que estuviese esperando el mismo mensaje procedente de una réplica del mismo objeto. Para lo cual fue necesario filtrar los mensajes por destinatario.

La gran cantidad de acciones que presenta el protocolo, de la práctica, detallado en la sección 3.3.2, también representó una complicación al momento del desarrollo, pues cada una de ellas significaba un esfuerzo muy significativo al momento de desarrollar. Por ejemplo, el tomar un vaso, aparte de la construcción de los objetos 3D del vaso y de la vitrina, es necesario programar las animaciones de las puertas de la vitrina, configurar la acción en el tutor y sus dependencias, programar los mensajes entre la puerta y el vaso, además de la acción de entregar el vaso al avatar. Esto agranda la cantidad de trabajo necesario para implementar un proceso largo.

Este último problema también se presentó al momento de realizar las pruebas, pues si se encontraba un error, era necesario volver a realizar toda la práctica después de corregirlo debido a los cambios de estado de los objetos.

Está previsto en un futuro cercano realizar pruebas del laboratorio virtual con profesores y alumnos (de grado y master) de biotecnología, y así, con sus experien-

cias, mejorar el entorno virtual.

Con posterioridad a las pruebas antes mencionadas, se van a implementar las otras fases de esta práctica de laboratorio y otras prácticas de biotecnología. Éstas deberían compartir los objetos 3D como el edificio o las herramientas, convirtiendo así el entorno en algo más parecido a un laboratorio real donde los avatares realicen experimentos diferentes en un mismo espacio.

Como trabajo futuro, considero que el tutor se puede complementar con un componente tutor externo programado en un lenguaje más potente, como el propuesto por Zayra Madrigal [Madrigal, 2010], pues *lsl* es muy limitado al no ser un lenguaje orientado a objetos. Este nuevo componente podría enviar mensajes de tutoría más inteligentes. Por ejemplo, le podría indicar al alumno la acción a realizar dependiendo de la zona en la que esté o si se encuentra parado sin hacer nada, así como podría dejar de brindar ayuda si el estudiante está realizando bien la práctica y reanudarla si comienza a equivocarse.

Cabe destacar que de este proyecto ha sido publicado en dos artículos. El primero, llamado “*An OpenSim-based Virtual Lab for biotechnology education*” [Ramírez et al., 2011], fue publicado en la revista “*IEEE Learning Technology*”. El segundo, llamado “*An architecture for virtual labs in engineering education*” [Rico et al., 2012], se publicó en la conferencia *Global Engineering Education Conference (EDU-CON)* de la IEEE. Este artículo se hizo merecedor del “*Best paper award*” en la categoría “*Infrastructure and Technologies for Engineering Education*”.

Bibliografía

- [Bartle, 2003] Bartle, R. (2003). *Designing Virtual Worlds*. New Riders Games.
- [Bartle, 2005] Bartle, R. (2005). *Virtual worlds: Why people play*. Charles River Media, Inc.
- [Bell, 2008] Bell, M. W. (2008). Toward a definition of virtual worlds. *Journal of Virtual Worlds Research*, 1:1–5.
- [Beyond Distance Research Alliance and GENIE, 2011] Beyond Distance Research Alliance and GENIE (2011). Second world immersive future teaching (swift). <http://www2.le.ac.uk/projects/swift>. [Acceso: 30-11-2011].
- [Bitmanagement Software GmbH,] Bitmanagement Software GmbH. Bs collaborate. http://www.bitmanagement.de/download/BS_Collaborate/BS_Collaborate_documentation.pdf. [Acceso: 27-01-2012].
- [Boulos et al., 2007] Boulos, M. N. K., Hetherington, L., and Wheeler, S. (2007). Second life: an overview of the potential of 3-d virtual worlds in medical and health education. *Health Information & Libraries Journal*, 24(4):233–245.
- [Ernstberger, 2009] Ernstberger, P. (2009). Linden dollar and virtual monetary policy. *Macroeconomics*.
- [Hrastinski, 2007] Hrastinski, S. (2007). *Participating in Synchronous Online Education*. PhD thesis, Department of Informatics, School of Economics and Management, Lund University.
- [Kaplan and Yankelovich, 2011] Kaplan, J. and Yankelovich, N. (2011). Open wonderland: An extensible virtual world architecture. *Internet Computing, IEEE*, 15(5):38–45.
- [Kumar et al., 2011] Kumar, S., Gankotiya, A. K., and Dutta, K. (2011). A comparative study of moodle with other e-learning systems. In *Electronics Computer Technology (ICECT), 2011 3rd International Conference on*, volume 5, pages 414–418.
- [Linden Research, Inc., 2011a] Linden Research, Inc. (2011a). Case study: The open university. http://wiki.secondlife.com/wiki/Case_Study:_The_Open_University. [Acceso: 28-11-2011].

- [Linden Research, Inc., 2011b] Linden Research, Inc. (2011b). History of second life. http://wiki.secondlife.com/wiki/History_of_Second_Life. [Acceso: 27-09-2011].
- [Linden Research, Inc., 2011c] Linden Research, Inc. (2011c). Second life education: The virtual learning advantage. [Acceso: 10-11-2011].
- [Linden Research, Inc., 2011d] Linden Research, Inc. (2011d). Second life education/get started. http://wiki.secondlife.com/wiki/Second_Life_Education_-_Get_Started. [Acceso: 10-11-2011].
- [Madrigal, 2010] Madrigal, Z. (2010). *Propuesta de Arquitectura para Incluir a las Simulaciones en los Sistemas de E-learning: Una Extensión al Modelo SCORM*. PhD thesis, Facultad de Informática, Universidad Politécnica de Madrid.
- [OpenSim Team, 2011a] OpenSim Team (2011a). History. <http://opensimulator.org/wiki/History>. [Acceso: 30-09-2011].
- [OpenSim Team, 2011b] OpenSim Team (2011b). What is opensimulator? http://opensimulator.org/wiki/Main_Page. [Acceso: 30-09-2011].
- [Petrakou, 2010] Petrakou, A. (2010). Interacting through avatars: Virtual worlds as a context for online education. *Comput. Educ.*, 54:1020–1027.
- [Ramírez et al., 2011] Ramírez, J., Rico, M., Berrocal, M., Riofrío, D., and De Antonio, A. (2011). An opensim-based virtual lab for biotechnology education. *IEEE Learning Technology*, 13(4):18–20.
- [Renneberg, 2008] Renneberg, R. (2008). *Biotechnología para principiantes*. Reverté.
- [Rico et al., 2011] Rico, M., Martínez-Muñoz, G., Alaman, X., Camacho, D., and Pulido, E. (2011). A programming experience of high school students in a virtual world platform. *International Journal of Engineering Education*, 27(1):52–60.
- [Rico et al., 2012] Rico, M., Ramirez, J., Riofrío, D., Berrocal-Lobo, M., and de Antonio, A. (2012). An architecture for virtual labs in engineering education. In *Global Engineering Education Conference (EDUCON), 2012 IEEE*, pages 1–5.
- [SJSU’s School of Library & Information Science, 2011a] SJSU’s School of Library & Information Science (2011a). Sloodle. http://slisweb.sjsu.edu/sl/index.php/Sloodle_Home_Page. [Acceso: 01-11-2011].
- [SJSU’s School of Library & Information Science, 2011b] SJSU’s School of Library & Information Science (2011b). Sloodleuserdocs. <http://slisweb.sjsu.edu/sl/index.php/SloodleUserDocs>. [Acceso: 01-11-2011].